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:

Dialogue Strategies for Multimedia Retrieval: Intertwining Abductive Reasoning and Dialogue Planning

Ulrich Thiel, Jon Atle Gulla, Adrian Müller and Adelheit Stein

Published in collaboration with the
British Computer Society

BCS

Springer

Dialogue Strategies for Multimedia Retrieval: Intertwining Abductive Reasoning and Dialogue Planning

Ulrich Thiel, Jon Atle Gulla, Adrian Müller, and Adelheit Stein
GMD–IPSI
Dolivostr. 15, D–64293 Darmstadt, Germany

Abstract

Multimedia information retrieval is an inherently interactive process. When the user enters a conceptual query, that is, a specification of some information need in terms of abstract concepts, there may be many alternative ways of interpreting the query. This, in turn, affects the determination of what items are relevant for retrieval. In the MIRACLE system, these problems are tackled by combining abductive reasoning and dialogue planning. Whereas MIRACLE's abductive retrieval engine is capable of deriving different interpretations of ambiguous queries, the dialogue planner employs a comprehensive conversational dialogue model for negotiating queries, clarifying information needs, and explaining retrieval results.

1 Introduction

Multimedia retrieval systems have to solve at least two different tasks: First, the relevant items have to be identified, and second, they have to be presented in such a way that the user can relate them to each other, and, what is often more complicated, to the query. In some situations, users might prefer to draw a sketch or to provide an example picture, e.g., when they are looking for similar objects, but are incapable or unwilling to provide a description. This is the assumption underlying most approaches to *content-based retrieval* (e.g., [26]), which exploit a similarity measure established between non-textual items like pictures to browse the database. As unrestricted browsing is highly ineffective in large databases, though, the need to impose a structure onto the search process arises. In most cases, we do not want to be dependent on isolated atomic features (such as a certain color) of the information item when navigating through the collection of items. The users prefer to think in terms of conceptualizations, for example, "sunset in the mountains", which comprise a number of features (colors, textures, illumination) together with semantic (conceptual) characterizations (sky, rocks, etc.). Hence, we have to construe the representation of a non-textual item as a structured entity or document.

Providing representations is the prerequisite for conceptual search. However, as is known to anyone trying to describe a non-textual object in their own words, there are many ways of doing this. Many ways of representing the contents of a multimedia document appear plausible, and there are many more ways of determining that a given object is relevant. Starting with a given conceptual query, we might use different strategies, e.g., employ different domain rules, to obtain (partially overlapping) alternative result sets. If the task of maintaining these access paths to the database can be left to the system, users can concentrate on their primary goal, i.e., assessing the relevance of items. As a matter of fact, the relevance assessment process can be supported by the system, for instance, by making explicit the retrieval rules that resulted in a certain object, showing alternative ways to handle a request, or negotiating a modification of an unsatisfactory query. These capabilities—among others—require a far more complex user interface than those currently in use. Furthermore, the demand for user guidance in multimedia retrieval applications causes a shift from exploratory interaction styles (browsing) to "conversational dialogue structures" [42].

Interfaces that act as intelligent mediators between users and other components of an information system aim at *exploiting the dialogue context* to make the interaction options, their meanings and their consequences transparent (cf. [22, 41]). This includes active context-based assistance in any phase of the interaction, as for example clarification of information needs and of related dialogue goals, or means for accomplishing these goals. Thus, the importance of monitoring the pragmatics and semantics of the dialogue is emphasized as well as the prominent role of supportive meta-communication. In sum, a complete solution to the problem of multimedia information retrieval has to address—at least—the following aspects:

- multimedia indexing combining feature extraction with semantic annotation,
- methods for bridging the gap between the users' conceptualization of information needs and the system's way of representing items, and
- dialogue structures and strategies that are appropriate in this situation.

In this article, we focus on the two latter issues; we introduce *an approach to handling ambiguous queries by intertwining query interpretation and flexible dialogue management*—employing abductive reasoning as a general inference technique in both areas (the indexing problem is treated in detail by Müller and Kutschekmanesch [25], this volume).

In the remainder of this article we will first provide an overview of related work in intelligent information retrieval (Section 2). In Section 3 we introduce MIRACLE, an experimental information retrieval system that combines multimedia indexing and retrieval components with a conversational dialogue manager. As abductive reasoning plays a key role for both the retrieval engine and the dialogue manager, we give a short account of this inference technique in Section 3.1 and discuss the basic features of the dialogue model and the dialogue manager in the Sections 3.2 and 3.3. Some examples taken from a MIRACLE dialogue session are discussed in Section 4 to illustrate our first experiences and lessons learned. The conclusions to the paper summarize the arguments brought forward and point out a number of open issues for future work and research in this area.

2 Related Work on Intelligent Information Retrieval

The process of finding information in large amounts of stored data involves a variety of reasoning tasks, ranging from problem definition to relevance assessment. Intelligent information retrieval (IIR) systems are intended to support the user in these tasks, and over the last decades a growing number of approaches to employing automatic reasoning techniques have been proposed.

The first approaches were influenced by experimental *question-answering systems* in the sixties and seventies (e.g., SIR, [32]). The main concern was to replace the document representation, which at that time was based on inverted files or term vectors, by more expressive alternatives: semantic networks and frames. Thus, the representation could be subjected to semantic analysis and could provide a basis for knowledge-based matching procedures.

Later, several projects on *intelligent information systems* have demonstrated the feasibility of extracting factual information from texts for the purpose of semantic indexing, extracting, and automatic creation of hyperlinks. In these systems, knowledge-based methods are combined with appropriate linguistic tools (such as "partial semantic parsers") to overcome the limits of statistical text processing. One of the most well-known systems in this category is General Electric's SCISOR [33]. This system can be used for information filtering and retrieval and also allows some new ways to handle information. For example, the "spontaneous retrieval" mechanism uses so far unanswered queries (instead of a user-edited profile as in traditional SDI) to notify the user when a document conveying relevant data is entered. Also, the idea of transforming background material into text knowledge bases has been explored. Whereas Simmons' approach [36] aims at providing a comprehensive knowledge base for expert systems, the CODER system [13] combines the filtering of incoming messages with retrieval operations in the textual background information base. Here, the user can use newly arrived messages as an entry point to the database. Similar techniques have been applied to business information by Carnegie Group Inc., resulting in several commercial prototypes that profitably employ state-of-the-art text understanding technology.

The extraction of information from incoming and background texts offers new ways to handle and access the texts. This requires, however, a new approach to the design of the user interface, since traditional query languages as well as simple hypertext browsing techniques cannot cope with the inherent complexity of the information available. This was illustrated by the TOPIC/TOPOGRAPHIC project at the University of Konstanz, Germany. TOPIC is an experimental text analysis system featuring a parser based on the conceptual knowledge of its domain. The parser is organized as a collection of distributed lexicalized grammar modules (word experts). It also uses linguistic knowledge on text grammar to identify discourse topics of a text or paragraph [17]. This enables the retrieval mechanism of TOPOGRAPHIC to select relevant documents, and to access relevant parts (extracted facts, passages, sections) with a high degree of precision. Furthermore, based on the representations of text fragments, the system can identify similar, contradicting or complementing information, enabling the user to follow semantic hyperlinks (cf. [18, 43]).

Whereas knowledge-based matching extends the potential for semantic retrieval, other AI techniques can be employed to achieve even more inferential power. The applicability of rule-based reasoning, which was demonstrated by the first working expert systems, stimulated a lot of experiments in IR (see [4] for an overview). Most of these efforts took the expert system approach literally and devised "automatic search intermediaries". Exploiting the semantic and—to some limited extent—the pragmatic knowledge about a specific problem domain, these systems were able to assist inexperienced users in query formulation and searching. For instance, the CANSEARCH system [30] incorporates a comprehensive domain model that is used to guide the user through a sequence of a priori defined choices represented as frames. Based on this dialogue, the system detects the relevant features of the user's problem, assigns appropriate search terms, and accesses the database. The PLEXUS system [46] features a similar functionality; however, since here natural language problem statements are the starting point of the dialogue, a component for semantic reasoning was added.

These prototypes are intentionally restricted to functions that simulate a human intermediary (the data are assumed to be stored in a traditional IR system), whereas other expert system designs aim at increasing the responsiveness and flexibility of the retrieval system. The I3R system [11, 12] is designed as an interface providing the usual intermediary functions as well as browsing options. Thus, the user is enabled to directly explore parts of the term–concept–document network, which is the basis of the retrieval process. The integrated probabilistic retrieval engine supporting relevance feedback and clustering was later replaced by a spreading activation mechanism [11] in order to retrieve documents by plausible inference as suggested by van Rijsbergen. His proposal was based on the assumption that "*... to design the next generation of IR systems, we will need to have a formal semantics for documents and queries. This semantic representation will interact with other types of knowledge in a controlled way, and this way is inference!*" ([34], p. 81). According to this proposal, in order to estimate the relevance of a document *D* with respect to a query *Q*, we have to estimate the probability $p(D \rightarrow Q)$. A model-theoretic procedure which can be used for this purpose is logical imaging [34]. Since imaging is independent of a specific notion of inference, different logics can be used in this framework. First prototypes show the viability of the approach (e.g., [10]).

The idea of *logic-based information retrieval* is more general than previous IR models, since it abstracts away from the inference mechanism, which can be probabilistic reasoning (as proposed by [34]) or some other logical procedure, as well as from the data representation format [27]. As a consequence, it encompasses a variety of approaches which go beyond the heavily domain-dependent rule-based reasoning of traditional expert systems. For instance, Watters and Shepherd employ resolution as a general deduction method to determine relevant documents, which are represented by bibliographic facts in a Prolog database [47]. Natural language queries are translated into a logical form that can be evaluated by the inference engine. Hess integrated linguistic processing and deductive retrieval to access relevant passages and sentences [19]. Other extensions aimed at integrating uncertainty into a logical framework (e.g., [15]).

Another approach combined the inferential process with a measure for semantic similarity: The RIME system [9] is designed to manage high-precision queries on a corpus of medical reports. The retrieval model maps natural language query statements to an internal tree representation of documents and radiographic images. The retrieval process of RIME is based on tree transformations. At each transformation step an uncertainty value is added to calculate the relevance of the—potentially matching—documents. It is important to note that the (transformed) trees code semantic properties.

Other approaches employ a probabilistic inferential model based on Bayesian dependence nets (e.g., [45]). The INQUERY retrieval system [8], which is based on this model, estimates the probability that the user's information need *I* (as expressed in the query *Q*) is satisfied by a document *D*, i.e., $p(I \mid D)$, by combining evidence along the one or more paths between query and document nodes.

To sum up, we conclude that most of the proposals to use reasoning in information retrieval are mainly based on more or less restricted forms of deductive inference in a first-order or probabilistic logic. However, since the consequence (the query) is known and we want to know the set of potential premises (the documents), inferential processes might be appropriate which allow us to find those premises. Both (probabilistic) abductive reasoning and Baysian networks can accomplish this (in fact, for propositional logic and discrete networks they have been proven to be equivalent, cf. [31]).

Most research in IR as well as IIR has concentrated on the task of processing a given query, while neglecting the fact that the quality of retrieval is heavily dependent on the interaction process that can be established between user and system. Some researchers did not assume a single shot retrieval process: this is reflected by concepts like *relevance feedback* (cf. [35]) and *retrieval as interaction* (for example, [3, 6, 12, 20, 28]). Whereas these approaches tried to explore the nature of retrieval dialogues and proposed interface designs to support the interaction, little work has been done on making explicit, or explaining, the system's decisions. This problem has primarily been treated in the context of expert systems (for example, [24]).

In the area of IR, mainstream research has focused on the ranking problem due to the assumption of one-dimensional relevance scales. Information scientists, however, have argued for multidimensional concepts of relevance (see [14] for an overview). A system design capturing the full complexity of "user relevance" is surely beyond the current technology. However, given an explicit representation of the relevance criteria which caused a document to be selected, a guided exploration of the retrieval result can be accomplished [44]. Using the main concepts associated with a retrieved item, other documents of the same class can be identified. Thus, it is possible to establish hyperlinks between them in accordance with the viewpoint selected by the user. Whereas this technique is based on the notion of a hypertext as a coherent non-linear *monologue* explored by the reader, other proposals regard it as a *dialogue*, where the system's task is to present nodes from a hyperbase that are meaningful continuations of the interaction. For instance, Whalen and Patrick [48] argue for a "conversational hypertext", where a user can access further nodes in the hypertext base by typing in natural language queries or comments. Techniques for the automatic indexing of large hyperbases are proposed by Osgood and Bareiss: the browsing interface to such an indexed hyperbase allows the user to select conversational links like "Refocusing, Causality, Comparison and Advice" ([29], p. 310).

3 A Framework for Conversational Multimedia IR

Content-based access to multimedia information, logic-based retrieval methods, and retrieval as interaction/dialogue form the three backbones of the framework proposed in this article. As pointed out in the previous section, none of these three issues is entirely new to the IR community and the research done in related areas. Within the large body of relevant literature, we find several approaches which deal with the same problems we face while taking a *user-centered perspective on intelligent multimedia retrieval*. Hence, we can build on several elaborate retrieval models, human-computer collaboration models, and AI techniques developed elsewhere. We only rarely find, however, approaches that focus on a *combination, or integration, of these issues and research areas*, in particular, in terms of application in complex systems. For example, elaborate computational models of discourse and human-computer collaboration have been developed in AI and HCI respectively, but they often neglect the specific problems related to information retrieval.

In IR the *retrieval as interaction* point of view has become widely acknowledged over the last years. Following, for example, Belkin and Vickery [6] and Ingwersen [20], we would also like to emphasize the dynamic aspects of information seeking processes. Users of (multimedia) IR systems tend to lack a well-defined information need and problem-solving plan as the interaction starts, meaning that their vague or ambiguous information needs gradually change as the dialogue develops. This often results in a number of *ambiguous user queries* that an intelligent retrieval system should be able to deal with (Sections 3.1 and 4 give an account of how we tackle this problem).

However, this is only part of the whole picture. Ambiguous and changing information needs/strategies over longer phases of the interaction demand an elaborate account of the dialogue structure (cf. Section 3.2). We assume that users will benefit from a flexible user guidance that provides useful information-seeking strategies while allowing for deviations in a natural way, for example, if an explanation is needed or the strategy has to be negotiated between system and user. This type of interaction resembles a conversation to a large extent—hence we will refer to it as *conversational information retrieval*.

Last but not least, it is important to note that 'ambiguous queries' are not necessarily identical with 'ambiguous information needs', and that both are by no means restricted to natural language dialogue; ambiguity also occurs—and probably even more so—in graphical and multimodal dialogue. In such a context, ambiguity arises from semantic and pragmatic design decisions: it may come along with simplified views of the actual semantic relationships (neglecting, for instance, time or some other context parameter needed to pinpoint a fact like a person–place relationship), or it results from functional particularities of interface devices, e.g., sliders which allow the user to control the system in a technical way but require some training, because the effects of a manipulation may be not as clearly distinguishable as desired. For instance, given sliders for controlling the hue and contrast values of pictures to be retrieved, how do you adjust them to retrieve snapshots taken in bright sunlight? As these and other problems at the interface level seem to be inherent to multimedia IR, the need to devise an interaction model capable of dealing with these intricacies arises. Resolving naturally occurring ambiguities, both with respect to query/retrieval aspects and to dialogue aspects, is the principle concern of our research presented here.

Before turning to the discussion of our approach in detail, we will now give a short account of our system prototype MIRACLE—MultImedia concept Retrieval bAsed on logiCal query Expansion[1] and its basic components:

[1] The current version of MIRACLE accesses an experimental (though large) database in the domain of art history, which consists of SGML documents (artist's biographies, reference articles, etc.), factual knowledge, and descriptions of thousands of pictures of works of arts. The system is implemented in C, Smalltalk and Prolog and runs on SystemV and BSD UNIX platforms. A World Wide Web (WWW) interface provides access for users and system developers.

- The *Indexer* (called MAGIC—Multimedia oriented Automatic Generation of Indices and Clusters) combines probabilistic text indexing with representation methods for pictures derived from content-based retrieval approaches (for details cf. Müller/ Kutschekmanesch, this volume).
- The *Abductive Retrieval Engine* works on a knowledge base comprising a semantic domain model, a model of the document/object structure, and the semantic counterparts (concept index) to the syntactic index terms assigned to the multimedia documents in the database (cf. Section 3.1).
- The *Dialogue Manager* mediates between the user and the retrieval engine. To achieve an appropriate amount of user guidance, this component relies on an explicit dialogue representation and a repository of dialogue acts (tactics) and strategies (cf. Sections 3.2 and 3.3).

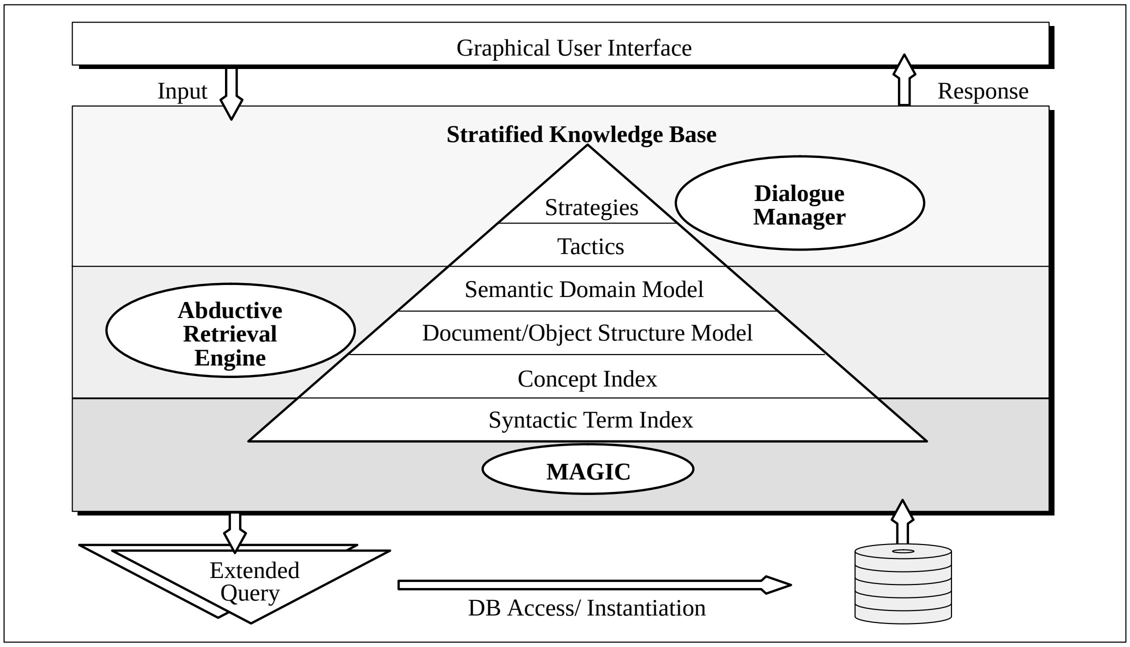


Figure 1: System architecture of MIRACLE

A user's query is a *description* of what the user is looking for and is unlikely to match database entries directly. To deal with these descriptions, we distinguish between an intensional (or: conceptual) representation of the domain, and the extensional model (i.e., instances retrieved from the database) of an inferred query interpretation. Applying abductive reasoning, we have devised a method for content-based concept retrieval in multimedia databases. Abduction can here be restricted to ground each proof (a *hypothesis*) on a special subset of available formulae only. A query is formulated at the intensional level and the inference mechanism generates query reformulations with respect to the available information structures, i.e., the inference process is set up to map from conceptual query statements to arbitrary information elements.

Distinguishing between the intensional and the extensional level demands a distinction between at least three global phases of the interaction, i.e., (1) query formulation; (2) inspection of the generated query interpretations; (3) inspection of instances retrieved from the database. Actual information retrieval dialogues, however, are often far more complex, and additional interaction options are to be included. For example, the user should be able to compare and evaluate the generated query interpretations before selecting the appropriate one to be executed; she might wish to ask questions (enter subdialogues for clarification), to withdraw or correct some previous decision, etc. To keep track of such complex interaction structures and to assist the user in finding an orientation, the dialogue manager must rely on an elaborate *model of dialogue* (cf. Section 3.2). Based on this model the dialogue manager dynamically builds up a *structured dialogue history* that is exploited to plan the subsequent dialogue steps (cf. Section 3.3).

3.1 Abductive Reasoning: An Inference Technique for Information Retrieval

Abduction is a form of inference apart from induction and deduction. The term ‘abduction’ was coined by C.S. Peirce. He defined abduction to be the explanation of the “surprising observation of a certain fact”. Talking about the nature of abductive inference in human reasoning, Peirce said in a 1903 lecture:
“The abductive inference comes to us like a flash. [...] it is the idea of putting together what we had never before dreamed of putting together which flashes the new suggestion before our contemplation.” ([7], p. 184)

Abductive reasoning tries to combine partial knowledge to form a more general concept of the world. This seems to be the most promising property of an abductive inference process. Many logic-based retrieval systems suffer from a kind of brittleness, if they are applied to real world problems and data. We suggest not to modify the logic theory alone, but to select an appropriate inference process, as well.

One of the crucial problems of using logic in IR is the need to model vague, and often even inconsistent properties of information by the formal and hence precise means of logical formalisms. A number of logic mechanisms have been developed which cope with uncertainty, default knowledge and the like. Unfortunately, the use of a calculus which allows the treatment of vague facts and rules does not automatically imply a higher degree of robustness. As a matter of fact, a logical theory will hardly cover all aspects of a real life domain. Thus, one needs to find an inference process which is able to recover from (or maintain) a partially inconsistent or insufficient model of the domain. In the last few years abduction has gained increasing interest in many fields of research in Artificial Intelligence. The most prominent applications can be found in diagnostic tasks, where the inference process needs to explain abnormal behavior in otherwise regular systems. This is due to the fact that abductive reasoning adds additional knowledge to a theory to relate isolated aspects of the problem under consideration.

Deduction		**Abduction**		
from:	a → b	*from:*	a → b	(rule)
	a		b	(observation)
infer:	b	*infer:*	a	(reason/hypothesis)
All human beings are mortal.		**All artists are famous.**		
Sokrates is a human being.		**Picasso is famous.**		
———		———		
Sokrates is mortal.		**(probably) Picasso is an artist.**		

A general characterization of abduction is: *abduction infers explanations* for a given observation. This can be illustrated by the basic inference step of abduction, which can roughly be described as a kind of inversion of Modus Ponens. Whereas Modus Ponens in deductive systems is based on material implication, abduction usually tries to find a *causal relationship* with respect to an observation and a given theory. But abduction need not be restricted to this kind of inference. Levesque [21] suggests not to demand a direct causal relationship between both formulae but to view the resulting formula as *one of the possible* reasonable explanations for the observation. When we use abduction within an information retrieval system, the relations are implicational and not necessarily causal. In general, abduction will find several possible explanations with respect to a given set of data and a query formulation. Reconsidering the need for robustness when using logic inference in IR, one should note that not all explanations need to be valid all together. Thus, we refer to each proof (explanation) as a feasible hypothesis.

An abductive system by definition generates all explanations for an observation with respect to a theory and a suitable form of logical implication. A logical theory $\mathcal{T}$ is defined over a language $\mathcal{L}$ of well-formed formulae, built from variables, constants and predicates. Given a theory $\mathcal{T}$ and a sentence a, which needs to be explained in terms of $\mathcal{T}$, an abductive reasoning process will yield a set of explanations (or hypotheses) H so that $\mathcal{T} \cup H \Rightarrow a$. This notion of abduction has two interesting properties:

- it can be implemented using a state-of-the-art meta-interpreter which is capable of tracing a logical calculus, and build a proof structure from the trace;
- since the meta-interpretation makes it possible to avoid accessing the database during the proof construction phase (it simply yields a list of propositions that remain to be proven independently), we have a means to cope with the incomplete or even inconsistent databases, which must be reckoned within the IR domain. We can

separate the intensional reasoning about the concepts from accessing the actual document representations (which can be regarded to form the extensional layer of the knowledge base).

In accordance with van Rijsbergen's formulation of the retrieval problem (see Section 2) we assume that a retrieval method attempts to prove that a multimedia document D entails (a part of) the query Q: $D \rightarrow Q$ [34]. By letting a user's query statement a to be $a = Q$ we can use the process of abductive reasoning to infer the user's information need directly from the given query statement. Abduction generates a set of explanations (document models plus additional hypotheses) which together imply the consequence (the query). Or, to give a more abstract reformulation of the process of abductive information retrieval:

> Assuming the given interpretation $\mathcal{T} \cup H$ for a to be true, the inference process proposes a way of understanding a query statement.

Furthermore, abductive reasoning will produce more than one reasonable query reformulation, since for non-trivial domains there will be—in most cases—more than one way of mapping a high-level information need to an aggregation of basic multimedia entities. Thus, the inference process also produces different possible readings of the query which may differ in their meaning on both the semantic and the structural levels. By maintaining the differences between these query interpretations, the dialogue manager of MIRACLE can provide *qualitative* feedback on how to process a given query statement.

The following example provides a brief introduction to sketch the technical details of the abductive inference process. Consider a database consisting of biographies of artists: each biography can be identified by the name of an artist; each artist has one or more *styles* of art, which can be modeled in terms of a traditional thesaurus. Additionally, the keywords from an art-and-artists thesaurus might occur somewhere in the textual components of the biographies. Now, consider a query statement like *Which artist is concerned with "Expressionism"?* The reasoning module infers at least two ways to interpret such a given query term: either as a keyword (e.g., by looking it up in the thesaurus index) or as a general full-text query phrase, i.e., a string as in "mad dogs and Englishmen". In the first case, the precision of the query reformulation will be higher (manual indexing of thesaurus terms), whereas the second case favors a higher recall (all textual occurences of the (stemmed) query terms). This distinction is inferred as an additional hypothesis, which will be frozen during the retrieval dialogue. The logical retrieval model will force all subsequent reformulations to comply with the interpretation chosen by the user.

As it would be beyond the scope of this paper to explain the details of the retrieval process employed in MIRACLE—in particular, the use of the probabilistic indexing component MAGIC—we refer to [25] (this volume) and to [44] for a detailed discussion. In the following we will concentrate on the interaction with the user, i.e., we will show a way in which the complex results provided by the abductive retrieval engine can be communicated. As we have sketched above, the inference process must take into account already established knowledge about the goals and plans of the user. In the example given above, the hypothesis should not be rejected in subsequent retrieval steps. The inferred knowledge is maintained as a set of constraints *C*, which prune the space of valid hypotheses. Thus, the definition below uses the constraints as filters.

> An explanation H , which yields $\mathcal{T} \cup H \Rightarrow a$, is valid iff $\mathcal{T} \cup H \cup C$ is consistent, where C (a set of constraints) is constructed from elements of $\mathcal{T}$.

Note that this definition might conflict with multiple hypotheses *H*. Since hypotheses can be mutually inconsistent, the inference process might find no consistent explanation, if the unstructured hypotheses are accumulated during a retrieval session. Thus, each individual retrieval strategy needs to be separated from the others in the dialogue management component, i.e., a practical system needs to maintain a collection of constraints which model the goals of each individual step in the retrieval dialogue. We will discuss this maintenance of constraints in the next sections.

3.2 The Dialogue Model: Conversational Roles and Dialogue Strategies

Human computer interaction plays a crucial role in information retrieval systems. Many users have initially vague or ambiguous information needs, and, accordingly, are unable or unwilling to formulate precise queries. Furthermore, as users' information needs and goals often change in the course of a retrieval session, they might need assistance in negotiating new tactics and strategies with the system.

The dialogue model we have developed and—partly—applied in previous retrieval systems (e.g., MERIT [39, 40]) has been enhanced and integrated into MIRACLE. The model comprises two parts that mutually constrain each other; hence, it allows the generation of a highly structured dialogue history:

- the *Conversational Roles model (COR)* is included to formally describe the general *local* interaction possibilities (dialogue moves and acts) of the two participants in any given dialogue situation;
- *dialogue scripts* are used as *global* plans to guide the user through the whole interaction. They describe action sequences that typically occur in information retrieval dialogues when a certain information-seeking strategy is adopted.

Guiding users through the global stages of the information retrieval dialogue by recommending appropriate problem-solving steps is helpful to users unfamiliar with the system. However, it is often impossible to anticipate all of the problematic situations that can occur during the interaction. To also cover unexpected situations and interactions, we allow the user to choose all of COR's local interaction options and afterwards try to interpret these options in light of the dialogue history. In this context, we understand an information retrieval dialogue as a cooperative negotiation between the two participants, human and computer, whose overall goals coincide. It is assumed that information seeker and provider aim to develop common plans based on the mutually accepted purpose of the current state and future direction of the dialogue [38].

COR is supposed to hold for all kinds of information-seeking dialogues: it covers the general "illocutionary aspects" of the dialogue contributions and of the changing dialogue roles of the two participants over time (cf. [37, 39, 40]). In this model, the basic units of dialogue are modeled as 'atomic' dialogue *acts*, i.e., the actual graphical or linguistic contributions of the participants (A and B). We distinguish 14 generic dialogue acts/moves that are categorized according to the main purpose ("illocutionary point") expressed. These generic dialogue acts include, for example, `request, offer, accept, reject-offer, inform, withdraw-request,` etc. Dialogue acts are elements of superordinated complex dialogue contributions, the *moves*, which are assigned the same illocutionary point as the respective atomic acts. The COR model of the entire *dialogue* is represented as a recursive transition network (Figure 2) consisting of dialogue states and transitions between the states, i.e., the moves.

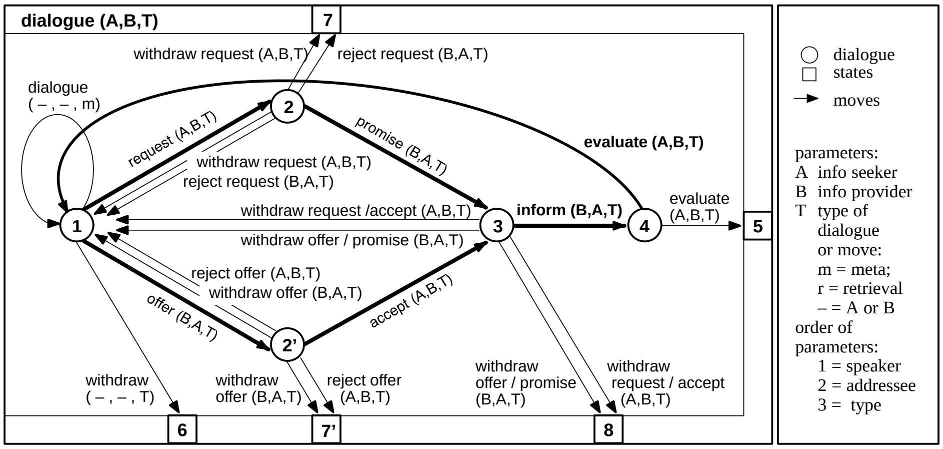


Figure 2: COR network for 'dialogue'

Sequences of moves starting in dialogue state 1 and returning to that state are called dialogue *cycles* (for example, `request` → `withdraw-request; offer` → `accept` → `inform` → `evaluate`). Bold arcs represent *expected* moves that comply with the role expectations; the sequences 1 – 2/2' – 3 – 4 – 1 are cycles that represent 'ideal' courses of action, where no withdrawals or rejections occur. Thus, for any of the dialogue states, the possible follow-up moves and possible action sequences can be described by the COR model. The initial move is either a `request` for information of the information seeker A or the information provider's (B's) `offer` to search for information and afterwards to present the retrieved items (inform move).

Moves are also represented as recursive transition networks (not displayed here). They can consist of atomic acts (e.g., `offer`), other moves (`assert` to supply context information), and embedded sub-dialogues (*dialogue* to solicit context information). As these subordinated moves and sub-dialogues are optional, a dialogue move can consist of a single atomic dialogue act, and even the entire move may be omitted in certain situations when the respective intention can be inferred from the context (for instance, a promise is often skipped in case the requested information can be given immediately, ε = empty move). The whole COR system is realized as a recursive transition network that at each step of the interaction offers a set of possible dialogue acts and accepts a number of acts that can change the state of the network.

The example dialogue on the next page, taken from a session with MIRACLE, allows us to illustrate how a *hierarchical dialogue history* is built up when traversing the recursive COR networks (the COR analysis is given in Figure 3). To simplify the presentation we give the example in natural language here—actually, the user enters queries in a special query form and performs all of the other dialogue acts by clicking on the respective buttons and icons (see Figures 6 and 8 in the next section). When the user asks for information about abstract art in Spain, the system finds the query ambiguous and suggests two different interpretations of it (see Figure 7 in Section 4). The user's query corresponds to a request, whereas the response from the system is an inform act. What happens later on in the dialogue is that the user first chooses interpretation 1 for the search, but right after withdraws this request and asks to use interpretation 2 instead.

For each dialogue act performed by the system or the user, the dialogue manager creates an entry in a dialogue history structure. This dialogue history stores all acts performed during the interaction, as well as queries and constraints associated with generated query interpretations. Acts following each other, like the request and the inform acts in the beginning of the dialogue, are inserted as siblings in the tree structure. An empty promise act is inserted between these two acts by the dialogue manager, since this is required by the COR model. As withdraw_request in the COR model is decomposed into a primitive withdraw act followed by a subdialogue, the subsequent three acts are inserted into the tree as a subdialogue under withdraw_request. When a subdialogue is initiated like that, it means that the interaction in this subordinate part is founded on what has happened at the higher level. Also, if the COR model is traversed so that the same kind of act is triggered several times at the same level of the dialogue, as the two request acts for query interpretations, we divide the dialogue into different *dialogue cycles*.

Fragment of a sample dialogue

U:	*Search for artists concerned with "abstract art" in "Spain".*	U: request
S:	*Here are some interpretations of your query ...* [presents interpretations]	S: inform
U:	*Use this interpretation in the search.* [chooses interpretation 1]	U: request
U:	*Stop, I made a mistake.* [interrupts the search]	U: withdraw_request
S:	*What do you want to do?* [presents dialogue options]	S: offer
U:	[Chooses to go back to the query interpretations]	U: accept
S:	[Presents other query interpretations]	S: inform
U:	*Use this interpretation in the search.* [chooses interpretation 2]	U: request
S:	[Shows artists and links to pictures]	S: inform
...		

An important aspect of the dialogue history is the *storing of constraints* connected to previously generated query interpretations. When a particular query interpretation is chosen, the hypothesis H underlying it is sent to the dialogue manager and stored together with the corresponding dialogue act in the tree. H_1 and H_2 in Figure 3 are the hypotheses corresponding to the two interpretations mentioned in the dialogue above and presented in Figure 7 below. These hypotheses reveal user decisions that have been taken at earlier steps and function as constraints on the system's interpretation of later queries. The complete dialogue history, thus, is a hierarchically structured analysis of what has happened at the dialogue act level and what decisions the user has made in the course of the interaction. As we will come back to in the next section, this is the only contextual information we need to deal with unexpected user actions and sequences of related user queries.

As opposed to the COR model, *dialogue scripts* give us structured guidelines for the information retrieval session. Based on a multi-dimensional classification of information-seeking strategies, the scripts proposed by Belkin et al. [5] implement prototypical interaction patterns corresponding to the various strategies. They are used as dialogue plans to guide the user through an IR session, recommending user acts and triggering appropriate system actions. Different from the case-based approach to dialogues adopted in a previous prototype (MERIT, cf. [40]), MIRACLE uses scripts

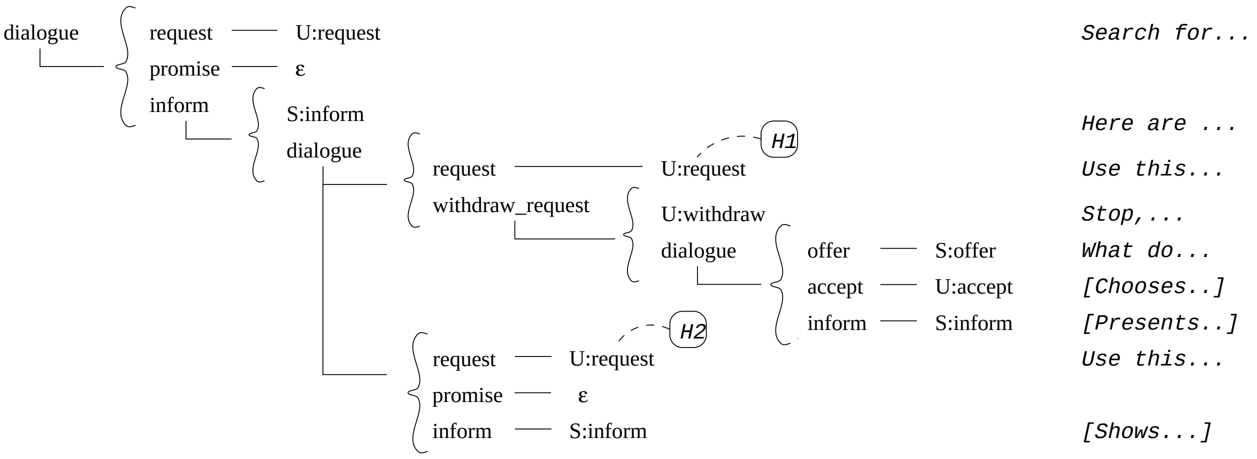


Figure 3: COR analysis of sample dialogue

that not only define straightforward paths, but also foresee branching points where the user may choose among a number of alternatives. Additionally, the scripts may contain some meta-sequences, e.g., for negotiating the further strategy or tactic in case the current strategy fails to fulfill the user's information need.

Consider the example script in Figure 4. On the left hand side of the figure, the stages and optional steps are enumerated; on the right hand side the possible follow-up steps. In state 6b, for example, the user can choose between reacting on the currently shown interpretation (one of the options under 7a) or going to the next interpretation (7b). In an introductory phase, which is common to all scripts, user and system negotiate the user's global information need until they come to an agreement (stages 1 – 4, not displayed here) about what to do. This enables the system to instantiate the relevant internal script, and the script proper continues in stage 5. The dialogue shown above started with a user's query corresponding to step 5 of this script.

5	U	*Generate query interpretations for ...* [query formulation]	→ 6
6	S	a. *Here's the one interpretation found ...*	→ 7a or 7a.2a
		b. *Here are some interpretations ...*	→ 7a or 7b
		c. *Sorry, no interpretation found, because ...* [explains]	→ 5 or 7c
		d. *Sorry, you have seen all alternatives.*	→ 5 or 7c
7	U	a. 1. *I like this.*	→ 9 or 7a.2a
		2. *Please explain the interpretation.*	→ 7a.2a or 7a.2b
	S	a. [explains]	→ 7a.3 or 7a.4/5
		b. *I can't explain, because ...* [explains]	→ 5 or 7c or 11
	U	3. *Please execute this query interpretation.*	→ **new script**
		4. *I don't like this.*	→ 1 or 7c
		5. *Let's quit.*	→ 11
	U	b. *Show me the next interpretation.*	→ ...
	S	c. *Here are some ways to find a good/better query interpretation ...* [list for choice]	→ ...
.....			
13	S	*Done. Goodbye.*	

Figure 4: Parts of a dialogue script

In general, scripts model recommended options for pursuing some information retrieval goal. Typical steps are specification of a query (by the user), generation of query interpretations and explanations (by the system), and inspection/evaluation of query interpretations or retrieved instances. Preconditions associated with the steps decide which alternatives the user should be presented and which actions the system is to perform. Postconditions describe the consequences of a particular move and will be explained in some more detail in the next section. Scripts, however, cannot predefine all alternative moves a user may wish to perform. The user might, for example, quit the dialogue before having retrieved anything, she may reject a system's offer, ask for help to fill out the query form or enter another clarification dialogue. In the dialogue example above, the user's withdrawal of her request is an example of a dialogue move that cannot be foreseen in the script. All these moves and subdialogues interrupt the interaction proposed by the script and call for some exceptional treatment. As they tend to express a desire to change the current interaction rather than a specification of some new interaction, special mechanisms for dealing with these cases are needed.

3.3 The Dialogue Manager

The dialogue manager administers the script being used and the COR analysis of the dialogue. Implemented as recursive state transition diagrams, the COR model and the scripts offer available acts at the various steps of the interaction and change their state in response to what acts were actually used. When an information retrieval session is initiated, an introductory script is chosen by the dialogue manager. This script prepares an empty dialogue history tree, since there has been no relevant interaction at that point, and it offers a set of recommended dialogue acts and an additional set of basic COR acts to the rest of the system. The recommended acts are accompanied by texts explaining how the acts relate to the overall task associated with the script, whereas the basic COR acts are offered as a means for deviating from the script in a way consistent with the dialogue context. The recommended acts are presented to the user as guidelines, and they make it easier for her to understand the flow of interaction. When the system or the user do a particular act, this act is sent to the dialogue manager, so that it can update the state of the script and add the act to the dialogue history. New sets of acts are then available, and this goes on until the whole session is completed. At each step of the interaction, thus, we tell the user what would be the natural thing to do, given her overall goal of the session. A complete session may involve the triggering of subscripts for specific parts, as well as interruptions leading to the termination or replacement of the current script.

In addition to guiding the user in this way, the dialogue manager is also responsible for

- providing the context (recorded in the dialogue history) for interpreting user queries, and
- interpreting user actions that are not modeled in the current script.

When new queries are entered or old queries are modified, we read these in light of the knowledge we already have obtained from the user. If the user chose a particular interpretation of a previous query, and the new one is partially dependent on that one, it is likely that she would stick to the constraints underlying the old interpretation also when it comes to the new one. For example, after receiving a huge list of artists at the end of the dialogue above, the user might try to restrict the search by adding a new search term:

U:	*Add to query that profession is "painter".*	U: request
S:	*Okay, I keep your chosen interpretation of "abstract art" and "Spain".*	S: promise
S:	*Here's the one interpretation found.*	S: inform

The dialogue history for these acts is shown in Figure 5. Since the new query is contextually dependent on the old one, the subdialogue is entered in the dialogue history as a subordinate to the previously chosen query interpretation.

When a query interpretation is chosen by the user, the constraint set corresponding to the interpretation is sent to the dialogue manager. The set is inserted together with the speech act into the dialogue history, like H_1 and H_2 in Figure 5, so that it can later be accessed by the query interpretation module. However, not all the constraint sets recorded in the dialogue history are considered relevant for later query interpretation. When a new query is to be interpreted, all relevant constraints have to be accumulated and sent to the retrieval module. This is done by including all constraints found in the cycle of the latest inserted speech act, as well as all constraints found in higher level cycles. When interpreting the last request in Figure 5, only H_2 is added as contraints to the abductive reasoning process (H_1 belongs to a cycle not superordinate to the latest inserted act). Consequently, we get only one interpretation of *"Spain"* (β_2 in Figure 7), since the system keeps the second interpretation of the previous query as a constraint C for interpreting the new extended query a^1: $\mathcal{T} \cup H \Rightarrow a^1$, where $\mathcal{T} \cup H \cup C$ is consistent and $C = H_2$. From the retrieval module's point of view, the dialogue manager is accessible through an abstract data type that adapts the reasoning process to the state of the dialogue and stores the whole interaction history.

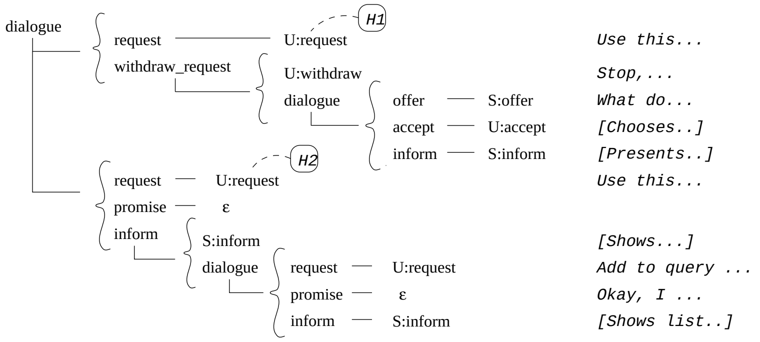


Figure 5: Dialogue history for extended query

Now, the other utilization of the dialogue history is connected to the notion of *unexpected dialogue moves*. Similarly to the approach taken in Verbmobil [1], these exceptional dialogue moves are detected by deviations from the dialogue plan (or, script). Even though the moves do not comply with the suggestions in the script, they are still assumed to follow the general interaction patterns modeled in COR, which provides more interaction options and hence more flexibility. The script is then terminated or suspended, and a set of *dialogue control rules* M are used to reformulate the unexpected move together with the moves in the dialogue history in terms of concrete system actions.

Consider the `withdrawal` act in the dialogue above, where the user is interrupting the use of a particular query interpretation in the search. As seen from the script in Figure 4, this move is not recommended to the user, but is taken from the general interaction options offered by the COR model (see Figure 2). The act is clearly a reaction against the script, though it is not obvious what the user wants to do instead. Choosing the other query interpretation and posting a new query are both consistent with such a `withdraw` act, and if some results have been obtained, there is also a choice as to whether these are to be stored or not.

When an unexpected act like this is detected, the system assembles all relevant acts from the dialogue history into a set D. It then tries to find hypotheses H about the user's concrete interaction wish, where H is given by the formula $H \cup M \Rightarrow D$. When several hypotheses are available, each is presented to the user, and she can determine what to do now by choosing one of them. In our withdrawal case, we use a rule in M stating that a wish to redo an old request is signalled by a following withdrawal act. As the `withdraw` act can be related to two `request` acts in the dialogue history, this rule gives rise to two different interpretations of the act:

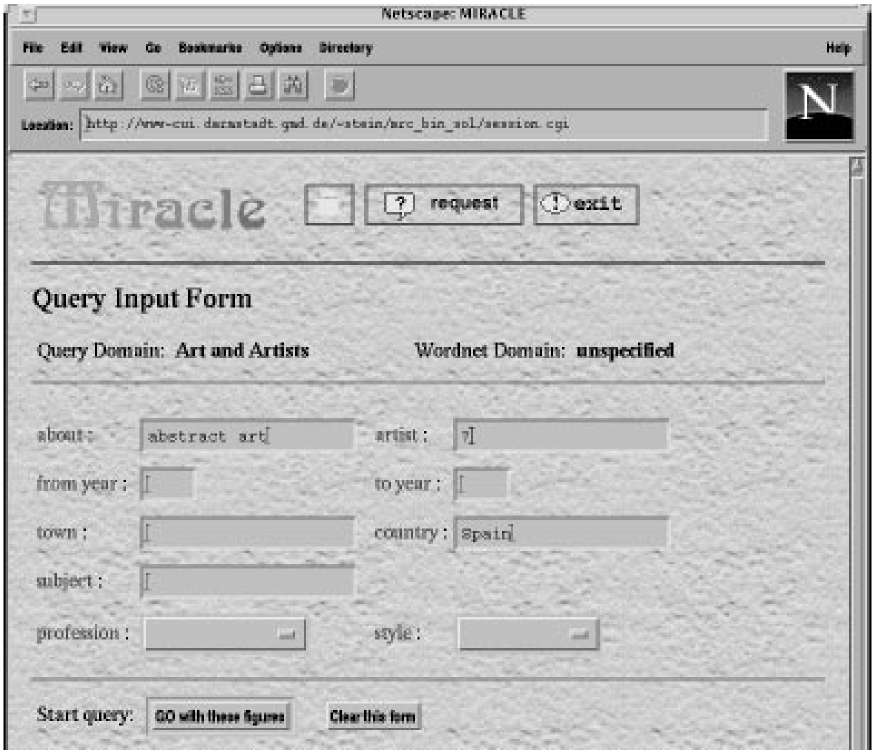
S:	*What do you want to do: post a new query or choose another query interpretation?*	S: offer
U:	*Give me another interpretation.*	U: accept

Just like in query interpretation, abduction is used to interpret the user's input and to find a more precise reformulation of it. Similar techniques of using the dialogue history to generate hypotheses about the user's intentions have been used earlier in text understanding systems (e.g., [23]) and explanation generators (e.g., [24]).

To integrate the dialogue model into MIRACLE, the repertoire of applicable scripts is being tailored to the retrieval functionality and the tasks that can be supported by the system. The dialogue control rules as well as the dialogue acts are formulated as logical formulas that can be included in abductive reasoning processes. Our method for constructing a dialogue history draws on another project using the COR model (cf. [2, 16]), though we have added the possibility of storing constraints and generating various views of the history. Since these constraints restrict the inferential possibilities of the retrieval engine, we can tune the system according to the context of the retrieval dialogue. For this integrated retrieval dialogue system, we also plan to extend the available interaction modes (direct manipulation and query forms) with natural language capabilities for generating explanations of the abduced results.

4 Working with MIRACLE

In the following we give a concrete example of a user's interaction with MIRACLE, that follows the sample dialogue and the script introduced in Section 3.2. We refer to the numbering of the stages and steps of this script. In the introductory sequence of our example the user selects a domain of interest (here, "art and artists"). The dialogue continues in stage 5 with the user's first query formulation: "*Search for* **artists** *concerned with* **abstract art** *in* **Spain**". Figure 6 shows this initial query as inserted by the user in MIRACLE's query form, whereas Figure 8 below displays the last step of our sample dialogue, the presentation of items retrieved from the domain database.



Location: http://www-cui.darmstadt.gmd.de/~stein/mrc_bin_sol/session.cgi

Query Input Form

Query Domain: **Art and Artists** Wordnet Domain: **unspecified**

about : abstract art artist : ?

from year : to year :

town : country : Spain

subject :

profession : style :

Start query:

Figure 6: A user's query

The abductive retrieval engine finds two interpretations of this query, thus the dialogue continues at 6.b, showing the first interpretation (β_1, cf. Figure 7). The formula is presented as a directed graph, directions indicating the inference sequence. For example, the left hand side of Figure 7(a) represents the following inference steps: *Assuming some artist A is textually relevant for "abstract art", if A is born in Spain, then he or she is a qualified artist.* The right hand side shows that this interpretation will evaluate the 'aboutness' statement by examining the textual components of the document collection. A, the missing link for the two parts of this query reformulation, is restricted to be a concept of type $artist()$ and thus it is a key for the document collection.

By selecting one of the rules in the first interpretation the user may ask for an explanation (stage 7a.2) of the corresponding concept. Selecting 'place of birth' the user gets informed (in an additional text window) that the query parameter $country(Spain)$ is mapped to the place of birth of the relevant artists. Since this differs from what she intended, she uses the 'next' button (stage 7b.1) to inspect alternative interpretations.

The system presents the second interpretation (β_2). Within this query interpretation, it is not the artist's place of birth, but the pictures' places of exhibition, which is restricted to "Spain". Here, the textual retrieval part (left hand side in Figure 7(b)) and the pictures are finally linked by the artist *A* via the computable relation $picture\_description()$.

After choosing interpretation β_2, the user asks the system to search the database. Technically speaking, MIRACLE is asked to find all *models* of the inferred formulae, which has been presented as interpretation β_2. This truth

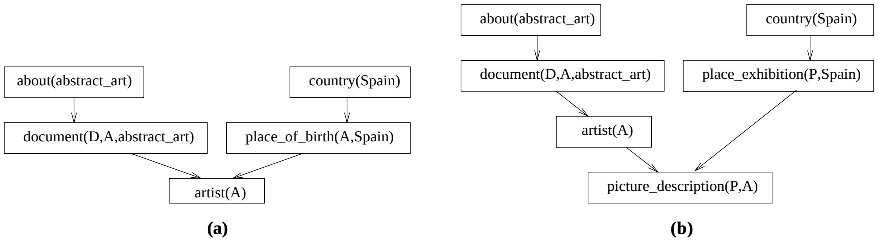


Figure 7: (a) First query interpretation (β_1); (b) Second query interpretation (β_2)

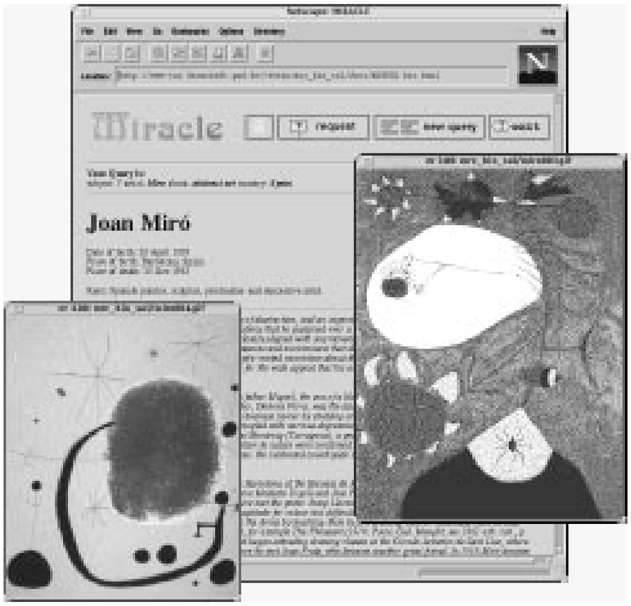


Figure 8: Biography and pictures retrieved

assignment is computed bottom-up, starting from basic predicates and recursively propagating potential instantiations to the root of the proof graph. Each unique model is returned as a *hit* to the user. For interpretation β_2, the number of records returned is annoyingly high, so the user decides to narrow the query (restrict the set of models) by inserting $profession(painter)$ (see query fields in Figure 6). This time, the system only considers the second interpretation of "Spain".

As the retrieval engine is told by the dialogue manager that the user wants to keep the second interpretation (from the previous dialogue step), it constrains the inference process so that the narrowed query is to be interpreted in the same manner as in β_2. The additional constraint is to filter the set of qualifying artists A by the condition $artist\_profession(A, painter)$. After computing the models, the system presents a list of artists with links to biographies and various pictures. Noticing Miró among these artists, the user finally adds *artist(Miró)* to the query and gets some information about the painter *Joan Miró*. As illustrated in Figure 8, there are both texts (biographies) and pictures associated with Miró in the database.

The example discussed shows how the functions of retrieval engine and dialogue manager are intertwined in the MIRACLE system. The dialogue planning is based on formal constraints which are defined on the semantic properties of the objects involved. For example, the association of a given retrieval result with one of the interpretations of the user's original query allows the explanation of the relevance decision of the system by referring to the assumptions underlying the interpretation.

5 Summary and Future Work

In this article we have introduced a theoretical framework for *intelligent conversational information retrieval* and its application in a multimedia information retrieval system, the MIRACLE prototype. Combining content-based information retrieval with a comprehensive dialogue model, the system is capable of assisting the user actively in her information seeking dialogue through all phases of the interaction. Both the logic-based retrieval engine and the dialogue manager employ abductive reasoning as the basic inference technique to resolve ambiguous information needs of the user and to plan cooperative system responses. The abductive retrieval engine of MIRACLE generates plausible interpretations of ambiguous user queries and offers these interpretations to the user for further negotiation. Employing a two-layered conversational dialogue model (COR and scripts) to dynamically build up a structured dialogue history, the dialogue manager analyzes this history and offers plausible continuations of the dialogue to the user. Thus, the system monitors and guides the interaction as the dialogue develops.

Using examples of the user–system interaction in MIRACLE, we discussed how the retrieval engine and the dialogue manager interact with each other to construct, depending on the user's input, a semantically and pragmatically coherent dialogue course. The retrieval engine and the dialogue manager interact through an abstract data type, which constrains the results of the inference process with respect to the current state of the dialogue and provides means for expressing and maintaining choice points of the dialogue history. As a side effect of raising a set of constraints, the search space of the retrieval engine shrinks to a reasonable size and the response times improve.

Most conceptual or logic-based retrieval systems addressing the problem of evaluating single queries rely on either rich representations, or on some modeltheoretic operations using an algebra featuring a similarity measure (e.g., [10, 27, 33]). In contrast to these systems, our approach focuses on combining a retrieval method taking into account the inherent ambiguity of conceptual queries (resulting from natural language formulations of information needs, changing or incomplete information needs, or—in the case of content-based retrieval—the need to define an operational equivalent to subjective notions like "bright colors") with active user assistance in *highly structured retrieval interaction processes.*

First experiences with test users of MIRACLE indicate that the offered assistance is well received by users with a complex—but not well-defined—information need. Other users, such as expert users of state-of-the-art retrieval systems, sometimes feel irritated by system features they are not used to and find these features somehow "cumbersome". A systematic evaluation, however, has not yet been performed and will be one of our next steps in the design–evaluation–redesign cycle.

Further open issues for our future work include: improving the interaction of the system components (i.e., retrieval engine, dialogue manager, and indexer), designing a more appropriate multimodal interface on the WWW to support the conversational approach, and incorporating a text generation component for generating explanations of the abductive retrieval method.

We are also experimenting with *semi-automatic indexing methods* for multimedia data-like images, where we will apply the techniques described in this paper. In this line of work, MIRACLE will infer access methods for non-intuitive domains (e.g. HTML data, feature extraction algorithms for digital images), given a high level query statement and a set of efficient index and retrieval algorithms, to ensure the effectivenes of the retrieval process. The intertwining of the dialogue manager and the retrieval engine will guide the user through a battery of non-intuitive query reformulating options.

References

[1] Alexandersson, J, Maier, E, Reithinger, N. A Robust and Efficient Three-Layered Dialogue Component for a Speech-to-Speech Translation System. In: Proc. of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL '95), Dublin, 1995, pp 188-193

[2] Bateman, J, Hagen, E, Stein, A. Dialogue Modeling for Speech Generation in Multimodal Information Systems. In: Proc. of the ESCA Workshop on Spoken Dialogue Systems—Theories and Applications (ETRW '95), Vigso, Denmark, 1995, pp 225-228

[3] Bates, M. The Design of Browsing and Berrypicking Techniques for the Online Search Interface. Online Review 1989; 13(5):407-424

[4] Belkin, NJ, Borgman, CL, Brooks, HM, et al. Distributed Expert-Based Information Systems: An Interdisciplinary Approach. Information Processing and Management 1987; 23(5):395-409

[5] Belkin, N.J, Cool, C, Stein, A, Thiel, U. Cases, Scripts, and Information Seeking Strategies: On the Design of Interactive Information Retrieval Systems. Expert Systems and Applications 1995; 9(3):379-395

[6] Belkin, NJ, Vickery, A. Interaction in Information Systems. The British Library, London, 1985

[7] Buchler, J (ed) Philosophical Writings of Peirce, Dover, New York, 1955

[8] Callan, JP, Croft, WB, Harding, SM. The INQUERY Retrieval System. In: Proc. of the 3rd International Conference on Database and Expert Systems Application. Valencia, Spain, Springer-Verlag, 1992, pp 78-83

[9] Chiaramella, Y, Nie, J. A Retrieval Model Based on an Extended Modal Logic and its Application to the RIME Experimental Approach. In: Vidick, JL (ed) Proc. of the SIGIR '90, ACM Press, New York, 1990

[10] Crestani, F, van Rijsbergen, CJ. Probability Kinematics in Information Retrieval. In: Fox, EA, Ingwersen, P, Fidel, R (eds) Proc. of the SIGIR '95, ACM Press, New York, 1995, pp 291-299

[11] Croft, WB, Lucia, TJ, Cringean, J, Willett, P. Retrieving Documents by Plausible Inference: an Experimental Study. Information Processing and Management 1989; 25(6):599-614

[12] Croft, WB, Thompson, RH. I3R: A New Approach to the Design of Document Retrieval Systems. Journal of the American Society for Information Science 1987; 38(6):389-404

[13] Fox, EA. Development of the CODER System: A Testbed for Artficial Intelligence Methods in Information Retrieval Information Processing and Management 1987; 23(2):341-366

[14] Froehlich, TJ. Relevance Reconsidered—Towards an Agenda for the 21st Century: Introduction to Special Topic Issue on Relevance Research. Journal of the American Society for Information Science 1994; 45(3):124-132

[15] Fuhr, N. Probabilistic Datalog—A Logic for Powerful Retrieval Methods. In: Fox, EA, Ingwersen, P, Fidel, R (eds) Proc. of the SIGIR '95, ACM Press, New York, 1995, pp 282-290

[16] Hagen, E, Stein, A. Automatic Generation of a Complex Dialogue History. In: Proc. of the Eleventh Canadian Conference on Artificial Intelligence (AI '96), Toronto, Canada, 1996

[17] Hahn, U. Topic Parsing: Accounting for Text Macro Structures in Full-text Analysis. Information Processsing and Management 1990; 26(1)

[18] Hammwöhner, R, Thiel, U. Content Oriented Relations between Text Units—A Strutural Model for Hypertext. In: Proc. of the HYPERTEXT '87, Chapel Hill, NC, pp 155-174

[19] Hess, M. An Incrementally Extensible Document Retrieval System Based on Linguistic and Logical Principles. In: Belkin, NJ, Ingwersen, P, Pejtersen, A (eds) Proc. of the SIGIR '92, ACM Press, New York, 1992, pp 190-197

[20] Ingwersen, P. Information Retrieval Interaction. Taylor Graham, London, 1992

[21] Levesque, H. A Knowledge-Level Account on Abduction. In: Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI '89), ACM, Press, New York, 1989, pp 1061-1067

[22] Maybury, MT (ed) Intelligent Multimedia Interfaces. AAAI Press/The MIT Press, Menlo Park, 1993

[23] McRoy, SW. Misunderstanding and the Negotiation of Meaning Using Abduction. Knowledge-Based Systems 1995; 8(2-3):126-134

[24] Moore, JD, Swartout, WR. A Reactive Approach to Explanation: Taking the User's Feedback into Account. In: Paris, CL, Swartout, WR, Mann, WC (eds) Natural Language Generation in Artifucial Intelligence and Computational Linguistics. Chapter 1, Kluwer Academic Publishers, 1991, pp 1-48

[25] Müller, A, Kutschekmanesch, S. Using Abductive Inference and Dynamic Indexing to Retrieve Multimedia SGML Documents. (this volume)

[26] Niblack, W, Faloutsus, C, Flickner, M, Petkovic, D, Equitz, W, Barber, R. Efficient and Effective Query by Image Content 1993; Research Report IBM, Computer Science, RJ 9453

[27] Nie, JY. Towards a Probabilistic Modal Logic for Semantic-based Information Retrieval. In: Belkin, NJ, Ingwersen, P, Pejtersen, A (eds) Proc. of the SIGIR '92, ACM Press, New York, 1992, pp 140-151

[28] Oddy, RN. Information Retrieval through Man-Machine-Dialogue. Journal of Documentation 1977; 33(1):1-14

[29] Osgood, R, Bareiss, R. Automated Index Generation for Constructing Large-scale Conversational Hypermedia Systems. In: Proc. of the 11th National Conference on Artificial Intelligence (AAAI '93). AAAI Press/The MIT Press, Menlo Park, 1993, pp 309-314

[30] Pollitt, S. CANSEARCH: An Expert System Approach to Document Retrieval. Information Processing and Management 1987; 23(2):119-138

[31] Poole, David. Probabilistic Horn Abduction and Bayesian Networks. Artificial Intelligence 1993; 64:81-129

[32] Raphael, B. SIR: A Computer Program for Semantic Information Retrieval. In: Minsky, ML (ed) Semantic Information Procesing. The MIT Press, Cambridge, MA/London, England, 1968, pp 33-145

[33] Rau, LF. Spontaneous Retrieval in a Conceptual Information System. In: Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI '89), ACM, Press, New York, 1989, pp 155-162

[34] van Rijsbergen, CJ. Towards an Information Logic. In: Belkin, NJ, van Rijsbergen, CJ (eds) Proc. of the SIGIR '89, ACM Press, New York, 1989, pp 77-86

[35] Rocchio, JJ. Relevance Feedback in Information Retrieval. In: Salton, G (ed) The Smart System—Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, New Jersey, 1971, Chapter 14

[36] Simmons RF. A Text Knowledge Base from the AI Handbook. Information Processing and Management 1987; 23(4):321-339

[37] Sitter, S, Stein, A. Modeling the Illocutionary Aspects of Information-Seeking Dialogues. Information Processing and Management 1992; 28(2):165-180

[38] Stein, A. Dialogstrategien für kooperative Informationssysteme: Ein komplexes Modell multimodaler Interaktion. Sprache und Datenverarbeitung 1995; (1):19-31

[39] Stein, A, Maier, E. Structuring Collaborative Information-Seeking Dialogues. Knowledge-Based Systems. Special Issue on Human-Computer Collaboration 1995; 8(2-3):82-93

[40] Stein, A, Thiel, U. A Conversational Model of Multimodal Interaction in Information Systems. In: Proc. of the 11th National Conference on Artificial Intelligence (AAAI '93). AAAI Press/The MIT Press, Menlo Park, 1993, pp 283-288

[41] Terveen, LG. Overview of Human-Computer Collaboration. Knowledge-Based Systems. Special Issue on Human-Computer Collaboration 1995; 8(2-3):67-81

[42] Thiel, U. Interaction in Hypermedia Systems: From Browsing to Conversation. In: Schuler, W, Hannemann, J, Streitz, N (eds) Designing User Interfaces for Hypermedia (Research Reports ESPRIT Project 6532 HIFI, Vol. 1). Springer-Verlag, Berlin, 1995, pp 43-54

[43] Thiel, U. Konversationale graphische Interaktion mit Informationssystemen: Ein sprechakttheoretischer Ansatz. Doctoral dissertation, University of Konstanz, Konstanz, Germany, 1990 (available from the author)

[44] Thiel, U, Müller, A. Why was this Item Retrieved? New Ways to Explore Retrieval Results. In: Agosti, M, Smeaton, A (ed) Information Retrieval and Hypertext. Kluwer, Boston, 1996 (forthcoming)

[45] Turtle, H, Croft, WB. Inference Networks for Document Retrieval. In: Vidick, JL (ed) Proc. of the SIGIR ’90, ACM Press, New York, 1990

[46] Vickery, A, Brooks, HM. PLEXUS—the Expert System for Referral. Information Processing and Management 1987; 23(2):99-117

[47] Watters, CR, Shepherd, MA. A Logic Basis for Information Retrieval. Information Processing and Management 1987; 23(2):433-445

[48] Whalen, T, Patrick, A. Conversational Hypertext: Information Access through Natural Language Dialogues with Computers. In: Bice, K, Lewis, C (eds) Proc. of the CHI ’88, ACM Press, New York, 1988, pp 289-292