An Object-Oriented Probabilistic Logic for Information Retrieval

Thomas Rölleke
University of Dortmund
roelleke@ls6.informatik.uni-dortmund.de

Abstract

This paper presents an information retrieval (IR) model based upon methods of object-oriented data modelling, probability theory, and logic.

The work is motivated by the growing need for integrating database (DB) systems and information retrieval systems. In particular requirements of IR in multimedia databases are considered.

The proposed model supports retrieval among several databases and allows for representing indexing knowlege, domain knowledge, and the knowledge of documents itself within the same framework.

1 Motivation and Background

The task of IR is the satisfaction of an information need. An information need is expressed via formulating a query. The formulation process should be supported by an effective user interface. And what does satisfy a query? The satisfaction of a query is given by a set of objects (documents) which are relevant to the query. For the purpose of determining the relevant objects IR systems need a knowledge representation. This knowledge representation must be capable to cope with the intrinsic uncertainty of knowledge and the vagueness of queries.

Nowadays there is a growing need for integrating DB systems and IR systems, since many commercial and technical applications have to store classical DB facts as well as text, pictures, and other media. In particular the multimedia character of data is getting very important. Thus searching for some information requires the ability to combine fact and document (reference) retrieval and to cope with heterogenous data. In addition, searching for information often requires to inspect more than just one database.

The IR model proposed in this paper takes into account those requirements. It follows the approach of [Rijsbergen 86] who proposes to describe the retrieval process as proving the implication $d \rightarrow q$, where d is the document representation and q is the query representation. An estimation of the probability $P(d \rightarrow q)$ could than serve as a relevance measurement.

From the above follows that an object-oriented data model for knowledge representation, which is combined with logical means and which can cope with probabilities to consider uncertainty might be a promising candidate to fulfill the mentioned requirements.

For combining object-orientated data modelling and logic means this work refers to ideas presented in [Kifer & Lausen 89]. For combining probabilities and reasoning about knowledge this work uses the semantic structure defined in [Fagin & Halpern 94].

2 Introduction and Aims

The first aim, of course, is to provide a knowledge representation for the knowledge derived by an indexing process. Example 1 shows the modelling of some indexing knowledge.

Example 1
```
d1:  document
0.9 d1[about → sailing]
0.8 d1[about → boats]
```

The meaning of the program given is as follows: The declaration `d1:document` defines object `d1` as a member of the class `document`. The probabilistic propositions `0.9 d1[about → sailing]` and `0.8 d1[about → boats]` express that `d1` is with a probability of `0.9 about sailing` and with a probability of `0.8 about boats`. `about` is called feature or attribute, the feature values `sailing` and `boats` are considered as objets.

The model also provides means to formulate rules. Example 2 expresses that a document is also about a term `T`, if the document is about a term `T'` which is a synonym of `T`. The second rule exploits the references of a document. A document is about some subject `S`, if the document refers to another document which is about the subject `S`.

Example 2
```
X[about → T] :- X: document, T: term, X[about → T'],
                T'[synonym → T]
X[about → S] :- X, Y: document, X[refers → Y],
                Y[about → S]
```

We refer to the left side of a rule as *rule-head* and to the right side of a rule as *rule-body*. The elements of a rule-body are separated by commas. An element is called subgoal.

The evaluation of a rule is based on intensional semantics. With the help of so-called event expressions we keep track of the proving process and thus we are able to cope with stochastical dependencies coming in by using the same proposition more than ones for deducing a further proposition. This is the same approach also used in [Fuhr 95b]. See [Fuhr & Rölleke 95] for a detailed description of using event expressions.

Example 3 shows how to pose a query.

Example 3
```
?- D: document, D[about → sailing], D[about → boats]
```

This query asks for documents about sailing and boats. Using rules any disjunction and conjunction of subgoals can be expressed.

The object-oriented architecture of the model allows for choosing arbitrary features for describing objects. Example 4 shows the adding of some author information.

Example 4
```
0.9 d1[about → sailing]
0.8 d1[author → peter]
```

The weight of the author value may reflect the responsibility of an author for a document.

3 Benefits

What do we gain from the introduced knowledge representation?

Using probabilistic propositions probabilistic knowledge derived by an indexing process can be expressed. As shown in the examples, the model provides thus means to cope with the intrinsic uncertainty of information retrieval and it also allows for modelling uncertain data in a database-like approach (see example 4). By the means of rules we can model the domain knowledge (e. g. thesaurus) we want to consider for indexing. Since rules gives us the capability to formulate complex queries, the model supports the combination of classical IR queries on *aboutness* with classical DB queries asking for attribute values (e. g. authors of a document). Thus it is easy to allow queries referring also to the logical structure and layout structure of documents as required in [Fuhr 95a]. Especially for multimedia databases such an open indexing structure as provided by this model is useful, because the many (and unpredictable) dimensions required for describing content and other knowledge about multimedia objects can be modelled by features and their values. Since feature values are taken to be objects in general instead of just terms, this approach supports non-standard indexing techniques required for other media than text. For example indexing with phonemes for speech retrieval or indexing with graphical objects for image retrieval is expressable in the proposed knowledge representation.

An import aspect of a knowledge representation for IR is its simplicity. We consider the proposed representation as quite intuitiv and straightforward, since probabilistic propositions and rules are the only syntactic elements of the model.

What is new considering this model?

The object-oriented architecture is a powerful and widely accepted knowledge representation. Assuming that we have several knowledge bases providing their knowledge in a way similar to the one described above, this model allows for combining those knowledge sources.

Example 5 shows the combination of two different knowledge bases.

Example 5

```
kb1, kb2:  knowledge-bases
0.9 kb1[d1:  document, 0.8 d1[about → sailing]]
0.7 kb2[d2:  document, 0.6 d2[about → sailing]]
```

The weights `0.9` and `0.7` reflect the reliability that the current context gives to the contexts `kb1` and `kb2`. The current context is referred to as *this-context*. A context is treated like an object, i. e. a context may also be described by features and their values. In addition, a context contains knowledge about other objects.

Example 6 shows how the presented structure allows for context-based retrieval and combining contexts.

Example 6

```
kb1[?- D: document, D[about → sailing]
0.8 d1]

this[?- D: document]

{kb1 ∨ kb2}[?- D: document, D[about → sailing]
0.72 d1
0.42 d2]
```

The first query gives an example for posing a query within a context. The second query is posed in the `this-context`. This query refers only to the knowlege of the current context. Thus the result is the empty set. The third query shows how the knowledge of several contexts is considered for processing a query. Contexts may be combined by disjunction ($\vee$) or conjunction ($\wedge$). Using disjuntion an object will be retrieved, if it is retrieved in at least one context. Using conjunction an object will only be retrieved, if it is retrieved in all contexts in the conjunction.

Besides the modelling of different knowledge sources contexts may also be used to build up user dependent contexts. The subjective domain knowledge and thus their subjective meaning of which documents should be retrieved could be considered within a context.

Having introduced the concept of contexts implies another idea. As described above a context bears knowledge about other objects. Extending this concept to documents allows for modelling the knowledge of the documents itself. Example 7 expresses that in document `d1` it is known that object `peter` is a member of the class `sailor`.

Example 7
`d1[peter: sailor, peter[birthday` $\rightarrow$ `22/11/88]]`

Furtheron the birthday of peter is mentioned in the document. This feature of the model goes beyond pure IR purposes. It allows for retrieval on objects described in documents. Of course, it is an open (linguistical) question, how to derive the knowledge of a document automatically. In the special case where an indexing process has access to the computer-based edited source of a text, it might be possible to exploit the format of the source. For example an indexing process for newspaper issues might exploit the format of the flat adverts to determine the features of flats. Thus users might query directly on the flats instead of querying for newspapers which contain adverts concerning flats with some wanted features.

4 Syntax and Semantics

The previous examples have introduced the syntax of the proposed language for knowledge representation.

Its main components are declarations, propositions, and rules. We call a set of these components program. Without weights and contexts the syntax is a subset of F-Logic ([Kifer & Lausen 89]. We can assign weights to declarations and propositions to express statistical knowledge. A program is encapsulated in a context and in a program itself we can assign reliablity factors for contexts.

For defining the interpretation of a program we use the semantic structure introduced in [Fagin & Halpern 94]. A structure is defined by a tuple $M = (S, K, \pi, P)$ where S is a set of possible words (also called states), K is a set of accessibility relations for several agents of the form (s, t), where $s, t \in S$, π is a truth value assignment with respect to a world, and P is a probability assignment, which assigns to each agent i and state s a probability space $P(i, s) = (S_{i,s}, \mu_{i,s})$. $S_{i,s}$ is the set of possible worlds which is considered by agent i in state s. $\mu_{i,s}$ is the probability function over these possible worlds. We define worlds to be disjoint and $\mu_{i,s}(S_{i,s}) = 1$.

Applying this semantics to the proposed IR model, we consider a context to be an agent who inspects a set of possible worlds. Figure 1 shows the set of possible worlds according to example 1 with a corresponding probability function.

s_1	0.9 * 0.8	d1[about → sailing], d1[about → boats]
s_2	0.9 * 0.2	d1[about → sailing]]
s_3	0.1 * 0.8	d1[about → boats]
s_4	0.1 * 0.2	

Figure 1: A possible worlds model

The given probability function over the possible worlds $s_1, \ldots, s_4$ assumes the propositions to be independent. Thus the sum over the possible worlds, where a formula is true, gives the meaning of the probability of every complex formula. We omit here the class declaration of object `d1` which is true in all possible worlds.

For clearness reason we should mention that the world s_1 represents the equivalence class of those worlds which assign the same truth values to the given declarations and propositions. The basic probability function on possible worlds may be equally distributed, since there is no reason why one world is more probable than another. This explains the two different interpretations of a weight of a world (equivalence class, respectively): 1. The weight is equal to the sum over the probabilities of the worlds which assign the same truth values. Referring to this interpretation we say that for example 80 percent of the worlds consider the formula `d1[about` $\rightarrow$ `boats]` being true. 2. For simplicity reason we consider the weight as the probability of a world.

So far we have shown the semantics for declarations and propositions in a given context. In example 5 we introduced means for modelling context-based knowledge. We investigate now two questions: What does a context program mean? Which semantics can be assigned to a combination of contexts in order to define the meaning of a query?

Figure 2 shows the formal definition of the semantics. The notation $(M, s) \models x$ expresses that the syntactic formula x is true due to a given model M with respect to a specific world s. The function $S_{obj-id,s}$ yields the set of all possible worlds which agent `obj-id` can reach from world s. The function $S_{obj-id,s}(x)$ yields the subset of $S_{obj-id,s}$ where x is true.

$$
\begin{array}{lcl}
(M,s) \models \text{proposition} & \Longleftrightarrow & \pi(s)(\text{proposition}) = true \\
(M,s) \models \varrho \text{ proposition} & \Longleftrightarrow & \varrho = \mu_{this,s}(S_{this,s}(\text{proposition})) \\
(M,s) \models \text{rule}_v & \Longleftrightarrow & (M,s) \models \text{rule-body}_v \Rightarrow (M,s) \models \text{rule-head}_v \\
(M,s) \models \text{rule-body}_v & \Longleftrightarrow & \text{for all subgoals: } (M,s) \models \text{subgoal}_v \\
(M,s) \models \varrho \text{ obj-id[program]} & \Longleftrightarrow & \varrho = \mu_{this,s}(\{s' | s' \in S_{this,s} \wedge \\
 & & \forall t \in S_{obj-id,s'}((M,t) \models \text{program})\})
\end{array}
$$

Figure 2: Formal semantics (propositions)

The first and second equivalence describe the semantics of a probabilistic proposition. The probability of a proposition is equal to the sum over the probabilities of the worlds, where the proposition is true. The semantics of a probabilitic declaration is defined analogously.

The third and fourth equivalence define the meaning of a rule and rule body. If a rule body is true with respect to a world s then the rule head is also true. The subscription v indicates the variable valuation. A rule body is true with respect to a world, if every subgoal is true.

The fifth equivalence describes the meaning of a program encapsulated in a context. The reliablity factor is equal to the sum of the probabilities of the worlds, from which agent (context) `obj-id` could reach a set of worlds, where the probabilistic program of the context is true.

For illustrating the semantics reconsider example 5. From 90 percent of the worlds of the current context agent `kb1` reaches a set of worlds, where the context program of `kb1` is true. Assuming that an agent reaches from a world in $S_{this,s}$ again the same set of worlds ($\forall t \in S_{this,s}(S_{this,t} = S_{this,s})$), figure 3 shows the semantics of the program of example 5.

s_1	0.9 * 0.7
s_2	0.9 * 0.3
s_3	0.1 * 0.7
s_4	0.1 * 0.3

t_1	0.8	d1[about $\rightarrow$ sailing]
t_2	0.2	

t'_1	0.6	d2[about $\rightarrow$ sailing]
t'_2	0.4	

Figure 3: Possible worlds model for example 5

Let the s-worlds be the worlds of the current context ($S_{this,s} = \{s_1, \ldots, s_4\}$). In 72 percent of these worlds both context programs are true, in 27 percent only the program of context `kb1` is true, and so forth. Agent `kb1` reaches from s_1 and s_2 the set of worlds $\{t_1, t_2\}$. In semantic terms: $S_{kb1,s_1} = S_{kb1,s_2} = \{t_1, t_2\}$ holds. For both worlds t_1 and t_2 holds the context program, since $S_{this,t_1} = S_{this,t_2} = \{t_1, t_2\}$.

We use now this semantics to define the meaning for combining contexts. Figure 4 shows the possible worlds distribution which defines the meaning of a probability of every complex formula according to a given combination of contexts.

For every possible conjunction of contexts, we get a subset of possible worlds. Each subset represents the possible conjunctions of the propositions of the considered contexts.

s_{11}	0.9 * 0.7 * 0.8 * 0.6	d1[about → sailing], d2[about → sailing]
s_{12}	0.9 * 0.7 * 0.8 * 0.4	d1[about → sailing]
s_{13}	0.9 * 0.7 * 0.2 * 0.6	d2[about → sailing]
s_{14}	0.9 * 0.7 * 0.2 * 0.4	
s_{21}	0.9 * 0.3 * 0.8	d1[about → sailing]
s_{22}	0.9 * 0.3 * 0.2	
s_{31}	0.1 * 0.7 * 0.6	d2[about → sailing]
s_{32}	0.1 * 0.7 * 0.4	
s_4	0.1 * 0.3	

Figure 4: Possible worlds model for combining contexts

An interesting question to look at is: Given two contexts, it might happen that every context defines a subjective probability for the same proposition. Example 8 shows a corresponding program.

Example 8

```
0.9 kb1[0.8 d1[about → sailing]]
0.7 kb2[0.6 d1[about → sailing]]
```

How do we define in this case the probability of the proposition `d1[about → sailing]` for a query combining the contexts? A disjunction of contexts yields the sum over all worlds, where at least one context is considered and the proposition is true. A conjunction yields the sum over all worlds, where all contexts are considered and in at least one context the proposition is true. See example 9 for the resulting weights of the queries.

Example 9

```
{kb1 ∨ kb2}[?- d1[about → sailing]
0.8376]

{kb1 ∧ kb2}[?- d1[about → sailing]
0.5796]
```

In the disjunction case the weight is computed by summing up the probabilities of the `kb1`-worlds where `d1[about → sailing]` is true and adding the probabilities of those worlds where the proposition is additionally true according to the consideration of context `kb2`. (Relating this to figure 4 and replacing `d2` by `d1` the probability 0.8376 is computed as the sum $\mu(\{s_{11}, s_{12}, s_{21}\}) + \mu(\{s_{13}, s_{31}\})$.) In the conjunction case the weight is equal to the sum of the probabilities of those worlds where both contexts are considered and the proposition is true in at least one context. $(0.5796 = \mu(\{s_{11}, s_{12}, s_{13}\}))$

5 Conclusion and Outlook

We have introduced an object-oriented knowledge representation which allows for reasoning and for modelling the intrinsic uncertainty of knowledge. Because of the object-oriented modelling arbitrary feature can be used to describe objects. Thus we achieve a data model suitable for multimedia IR purposes. The model also allows for storing DB facts and thus it provides a framework for integrating DB systems and IR systems.

Further work will be done on the following topics:

How do we define negation? An interesting point for investigation is the context-based definition of closed world assumption and open world assumption, respectively.

What are the advantages and disadvantages of using Dempster-Shafer theory instead of probability theory?

Do we need uncertain rules? They would increase the expressiveness but the evaluation is because of incoming stochastic conditions difficult.

Furtheron we would like to cope with a special case of dependent proposition, namely disjoint propositions. This would allow for modelling uncertain DB facts.

References

Fagin, R.; Halpern, J. (1994). Reasoning About Knowledge and Probability. *Journal of the ACM 41(2)*, pages 340–367.

Fuhr, N.; Rölleke, T. (1995). *A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems*. (Submitted for publication).

Fuhr, N. (1995a). Logical and Conceptual Models for the Integration of Information Retrieval and Database Systems. In: *East/West Database Workshop, Klagenfurt 1994*, pages 206–218. Springer, Berlin et al.

Fuhr, N. (1995b). Probabilistic Datalog - a Logic for Powerful Retrieval Methods. In: *Proceedings SIGIR '95*, pages 282–290. ACM, New York.

Kifer, F.; Lausen, G. (1989). F-Logic: A Higher-Order Language for Reasoning About Objects, Inheritance, and Scheme. In: *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 134–146. ACM, New York.

van Rijsbergen, C. J. (1986). A Non-Classical Logic for Information Retrieval. *The Computer Journal 29(6)*, pages 481–485.