IR Models based on Predicate Logic

Norbert Fuhr
The logical view on IR

- IR as Inference
- IR as uncertain inference
- Propositional vs. Predicate Logic
  - Disjoint events
  - Relational Bayes
  - Probabilistic rules
The logical view on IR
IR as inference

$q$ - query
$d$ – document

retrieval:
search for documents which imply the query: $d \rightarrow q$

Example: classical IR:
\[
d = \{t_1, t_2, t_3\}
\]
\[
q = \{t_1, t_3\}
\]
retrieval: $q \subset d$ ?

logical view:
\[
d = t_1 \land t_2 \land t_3
\]
\[
q = t_1 \land t_3
\]
retrieval: $d \rightarrow q$ ?
advantage of inference-based approach:
step from term-based to knowledge-based retrieval

e.g. easy incorporation of additional knowledge

d: 'squares'
q: 'rectangles'

example:

thesaurus: 'squares' $\rightarrow$ 'rectangles'

$\Rightarrow$: $d \rightarrow q$
IR as uncertain inference

d: 'quadrangles’
q: 'rectangles’
⇒ uncertain knowledge required
'quadrangles’ \( \Rightarrow 0.3 \) 'rectangles’

[Rijsbergen 86]:
IR as uncertain inference
Retrieval ˆ
estimate probability \( P(d \rightarrow q) = P(q|d) \)
Limitations of propositional logic:

conventional indexing (based on propositional logic): \( d = \{ \text{tree, house} \} \)

query: *Is there a picture with a tree on the left of the house?*
\( \Rightarrow \) query cannot be expressed in propositional logic

predicate logic:

\[
d: \text{tree}(t1). \text{house}(h1). \text{left}(h1,t1). \\
?- \text{tree}(X) \& \text{house}(Y) \& \text{left}(X,Y).
\]
Relational Structures: Datalog

Datalog program: finite set of rules each expressing a conjunctive query

\[ t(X_1, ..., X_k) : \neg r_1(U_{11}, ..., U_{1m_1}), ..., \neg r_n(U_{n1}, ..., U_{nm_n}) \]

where each variable \( X_i \) occurs in the body of the rule (this way, every rule is safe).

\begin{align*}
\text{woman}(X) & : \text{person}(X), \text{sex}(X, \text{female}). \\
\text{path}(X,Y) & : \text{link}(X,Y). \\
\text{path}(X,Z) & : \neg \text{link}(X,Y), \text{path}(Y,Z).
\end{align*}
Datalog rules

t(X_1, ..., X_k) : \neg r_1(U_{11}, \ldots, U_{1m_1}), \ldots, r_n(U_{n1}, \ldots, U_{nm_n})

corresponds to the logical formula

\forall X_1 \ldots \forall X_k \forall U_{11} \ldots \forall U_{nm_n}

\quad t(X_1, ..., X_k) \land \neg r_1(U_{11}, \ldots, U_{1m_1}) \land \ldots \land \neg r_n(U_{n1}, \ldots, U_{nm_n})

t(X_1, ..., x_k) is called the head and
r_1(U_{11}, \ldots, U_{1m_1}), \ldots, r_n(U_{n1}, \ldots, U_{nm_n}) the body.
A formula without a head is also called a fact
Datalog Properties

- horn predicate logic
- no functions
- restricted forms of negation allowed

\[ t(X_1, \ldots, X_k) : \neg r_1(U_{11}, \ldots, U_{1m_1}), \ldots, \neg r_n(U_{n1}, \ldots, U_{nm_n}) \]

- rules may be recursive (head predicate may occur in the body)

\[ r(X, Y) : \neg l(X, Z), r(Z, Y) \]

- sound and complete evaluation algorithms
IR Models based on Predicate Logic
The logical view on IR
Propositional vs. Predicate Logic

IR and Databases
The Logic View

Retrieval

- DB: given query $q$, find objects $o$ with $o \rightarrow q$
- IR: given query $q$, find documents $d$ with high values of $P(d \rightarrow q)$
- DB is a special case of IR!
  (in a certain sense)

This section: Focusing on the logic view

- Inference
- Vague predicates
- Query language expressiveness
IR and Databases
The Logic View

Retrieval

- DB: given query $q$, find objects $o$ with $o \rightarrow q$
- IR: given query $q$, find documents $d$ with high values of $P(d \rightarrow q)$
- DB is a special case of IR! (in a certain sense)

This section: Focusing on the logic view

- Inference
- Vague predicates
- Query language expressiveness
Inference

- IR with the Relational Model
- The Probabilistic Relational Model
- Interpretation of probabilistic weights
- Extensions
  - Disjoint events
  - Relational Bayes
  - Probabilistic rules
### Relational Model

**Projection**

<table>
<thead>
<tr>
<th>DOCNO</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ir</td>
</tr>
<tr>
<td>1</td>
<td>db</td>
</tr>
<tr>
<td>2</td>
<td>ir</td>
</tr>
<tr>
<td>3</td>
<td>db</td>
</tr>
<tr>
<td>3</td>
<td>oop</td>
</tr>
<tr>
<td>4</td>
<td>ir</td>
</tr>
<tr>
<td>4</td>
<td>ai</td>
</tr>
<tr>
<td>5</td>
<td>db</td>
</tr>
<tr>
<td>5</td>
<td>oop</td>
</tr>
</tbody>
</table>

**Projection:** what is the collection about?

\[
\text{topic}(T) :\neg \text{index}(D,T).
\]
Selection: which documents are about IR?

aboutir(D) :- index(D,ir).
Join: who writes about IR?

\[ \text{iraauthor}(A) : \text{index}(D, \text{ir}) \land \text{author}(D, A). \]
Relational Model

Union

<table>
<thead>
<tr>
<th>DOCNO</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ir</td>
</tr>
<tr>
<td>1</td>
<td>db</td>
</tr>
<tr>
<td>2</td>
<td>ir</td>
</tr>
<tr>
<td>3</td>
<td>db</td>
</tr>
<tr>
<td>3</td>
<td>oop</td>
</tr>
<tr>
<td>4</td>
<td>ir</td>
</tr>
<tr>
<td>4</td>
<td>ai</td>
</tr>
<tr>
<td>5</td>
<td>db</td>
</tr>
<tr>
<td>5</td>
<td>oop</td>
</tr>
</tbody>
</table>

\underline{Union}: which documents are about IR or DB?

\texttt{irordb(D) :- index(D,ir).}
\texttt{irordb(D) :- index(D,db).}
IR Models based on Predicate Logic
Inference
IR with the Relational Model

Relational Model
Difference

<table>
<thead>
<tr>
<th>DOCNO</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ir</td>
</tr>
<tr>
<td>1</td>
<td>db</td>
</tr>
<tr>
<td>2</td>
<td>ir</td>
</tr>
<tr>
<td>3</td>
<td>db</td>
</tr>
<tr>
<td>3</td>
<td>oop</td>
</tr>
<tr>
<td>4</td>
<td>ir</td>
</tr>
<tr>
<td>4</td>
<td>ai</td>
</tr>
<tr>
<td>5</td>
<td>db</td>
</tr>
<tr>
<td>5</td>
<td>oop</td>
</tr>
</tbody>
</table>

**Difference**: which documents are about IR, but not DB?

\[ \text{irnotdb}(D) \leftarrow \text{index}(D, \text{ir}) \& \neg \text{index}(D, \text{db}). \]
**IR Models based on Predicate Logic**

Inference

The Probabilistic Relational Model

### The Probabilistic Relational Model

[Fuhr & Roelleke 97] [Suciu et al 11]

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>DOCNO</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>IR</td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>DB</td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>IR</td>
</tr>
<tr>
<td>0.5</td>
<td>3</td>
<td>DB</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>OOP</td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>IR</td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td>AI</td>
</tr>
<tr>
<td>0.8</td>
<td>5</td>
<td>DB</td>
</tr>
<tr>
<td>0.3</td>
<td>5</td>
<td>OOP</td>
</tr>
</tbody>
</table>
The Probabilistic Relational Model

[Fuhr & Roelleke 97] [Suciu et al 11]

index

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>DOCNO</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>IR</td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>DB</td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>IR</td>
</tr>
<tr>
<td>0.5</td>
<td>3</td>
<td>DB</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>OOP</td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>IR</td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td>AI</td>
</tr>
<tr>
<td>0.8</td>
<td>5</td>
<td>DB</td>
</tr>
<tr>
<td>0.3</td>
<td>5</td>
<td>OOP</td>
</tr>
</tbody>
</table>

Which documents are about DB?
aboutdb(D) :- index(D,db).
The Probabilistic Relational Model

[Fuhr & Roelleke 97] [Suciu et al 11]

index

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>DOCNO</th>
<th>TERM</th>
<th>aboutdb</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>IR</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>DB</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>IR</td>
<td>0.7</td>
</tr>
<tr>
<td>0.5</td>
<td>3</td>
<td>DB</td>
<td>0.5</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>OOP</td>
<td>0.5</td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>IR</td>
<td>0.8</td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td>AI</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>5</td>
<td>DB</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>5</td>
<td>OOP</td>
<td></td>
</tr>
</tbody>
</table>

Which documents are about DB?

aboutdb(D) :- index(D,db).
IR Models based on Predicate Logic
Inference
The Probabilistic Relational Model

The Probabilistic Relational Model

[Fuhr & Roelleke 97] [Suciu et al 11]

<table>
<thead>
<tr>
<th>β</th>
<th>DOCNO</th>
<th>TERM</th>
<th>aboutdb</th>
<th>aboutirdb</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>IR</td>
<td>0.7</td>
<td>0.8*0.7</td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>DB</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>IR</td>
<td>0.5</td>
<td>3</td>
</tr>
<tr>
<td>0.5</td>
<td>3</td>
<td>DB</td>
<td>0.8</td>
<td>5</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>OOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>IR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td>AI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>5</td>
<td>DB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>5</td>
<td>OOP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which documents are about DB?
aboutdb(D) :- index(D,db).

Which documents are about IR and DB?
aboutirdb(D) :- index(D,ir) & index(D,db).
### Extensional vs. intensional semantics

**docterm**

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>DOC</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>d1</td>
<td>ir</td>
</tr>
<tr>
<td>0.5</td>
<td>d1</td>
<td>db</td>
</tr>
</tbody>
</table>

**link**

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>d2</td>
<td>d1</td>
</tr>
</tbody>
</table>

\[
\text{about}(D,T) :- \text{docTerm}(D,T).
\]

\[
\text{about}(D,T) :- \text{link}(D,D1) \land \text{about}(D1,T)
\]

\[
\text{q}(D) :- \text{about}(D,\text{ir}) \land \text{about}(D,\text{db}).
\]

**Extensional semantics:**

weight of derived fact as function of weights of subgoals

\[
P(q(d2)) = P(\text{about}(d2,\text{ir})) \cdot P(\text{about}(d2,\text{db})) = (0.7 \cdot 0.9) \cdot (0.7 \cdot 0.5)
\]

**Problem**

“improper treatment of correlated sources of evidence” [Pearl 88]  
→ extensional semantics only correct for tree-shaped inference structures
Extensional vs. intensional semantics

<table>
<thead>
<tr>
<th>docterm</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7</td>
</tr>
<tr>
<td>d1</td>
<td>d2</td>
</tr>
<tr>
<td>ir</td>
<td>T</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>d1</td>
<td>d1</td>
</tr>
<tr>
<td>db</td>
<td></td>
</tr>
</tbody>
</table>

about(D,T) :- docTerm(D,T).
about(D,T) :- link(D,D1) & about(D1,T)
q(D) :- about(D,ir) & about(D,db).

**Extensional semantics:**
weight of derived fact as function of weights of subgoals

\[
P(q(d2)) = P(about(d2,ir)) \cdot P(about(d2,db)) = (0.7 \cdot 0.9) \cdot (0.7 \cdot 0.5)
\]

**Problem**

“improper treatment of correlated sources of evidence” [Pearl 88] → extensional semantics only correct for tree-shaped inference structures
IR Models based on Predicate Logic

Inference

The Probabilistic Relational Model

Extensional vs. intensional semantics

docterm

\[
\begin{array}{ccc}
\beta & DOC & TERM \\
0.9 & d1 & ir \\
0.5 & d1 & db \\
\end{array}
\]

link

\[
\begin{array}{ccc}
\beta & S & T \\
0.7 & d2 & d1 \\
\end{array}
\]

about(D,T) :- docTerm(D,T).

about(D,T) :- link(D,D1) & about(D1,T)

q(D) :- about(D,ir) & about(D,db).

Extensional semantics:

weight of derived fact as function of weights of subgoals

\[
P(q(d2)) = P(about(d2,ir)) \cdot P(about(d2,db)) = (0.7 \cdot 0.9) \cdot (0.7 \cdot 0.5)
\]

Problem

“improper treatment of correlated sources of evidence” [Pearl 88]

→ extensional semantics only correct for tree-shaped inference structures
**Intensional semantics**

weight of derived fact as function of weights of underlying ground facts

**Method**: Event keys and event expressions

<table>
<thead>
<tr>
<th>β</th>
<th>κ</th>
<th>DOC</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>dT(d1,ir)</td>
<td>d1</td>
<td>ir</td>
</tr>
<tr>
<td>0.5</td>
<td>dT(d1,db)</td>
<td>d1</td>
<td>db</td>
</tr>
</tbody>
</table>

?- docTerm(D,ir) & docTerm(D,db).

gives

d1 \[dT(d1,ir) \& dT(d1,db)\] \[0.9 \cdot 0.5 = 0.45\]

<table>
<thead>
<tr>
<th>link</th>
<th>β</th>
<th>κ</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>l(d2,d1)</td>
<td>0.7</td>
<td>d2</td>
<td>d1</td>
<td></td>
</tr>
</tbody>
</table>
Intensional semantics

weight of derived fact as function of weights of underlying ground facts

**Method:** Event keys and event expressions

\[
\beta \quad \kappa \\
0.9 \quad dT(d1,ir) \quad d1 \quad ir \\
0.5 \quad dT(d1,db) \quad d1 \quad db
\]

?- docTerm(D,ir) & docTerm(D,db).

gives

\[
d1 \quad [dT(d1,ir) \& dT(d1,db)]
\]

\[
0.9 \cdot 0.5 = 0.45
\]
Intensional semantics

weight of derived fact as function of weights of underlying ground facts

**Method:** Event keys and event expressions

docterm

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\kappa$</th>
<th>DOC</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>$dT(d1,\text{ir})$</td>
<td>$d1$</td>
<td>$\text{ir}$</td>
</tr>
<tr>
<td>0.5</td>
<td>$dT(d1,\text{db})$</td>
<td>$d1$</td>
<td>$\text{db}$</td>
</tr>
</tbody>
</table>

?- docTerm(D,ir) & docTerm(D,db).

gives

d1 $[dT(d1,\text{ir}) \& dT(d1,\text{db})]$  

$0.9 \cdot 0.5 = 0.45$
Intensional semantics

weight of derived fact as function of weights of underlying ground facts

**Method:** Event keys and event expressions

<table>
<thead>
<tr>
<th>β</th>
<th>κ</th>
<th>DOC</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>dT(d1,ir)</td>
<td>d1</td>
<td>ir</td>
</tr>
<tr>
<td>0.5</td>
<td>dT(d1,db)</td>
<td>d1</td>
<td>db</td>
</tr>
</tbody>
</table>

?- docTerm(D,ir) & docTerm(D,db).

gives

d1 [dT(d1,ir) & dT(d1,db)]

\[
0.9 \cdot 0.5 = 0.45
\]
**Event keys and event expressions**

<table>
<thead>
<tr>
<th>docterm</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta )</td>
<td>( \kappa )</td>
</tr>
<tr>
<td>0.9</td>
<td>( dT(d1,ir) )</td>
</tr>
<tr>
<td>0.5</td>
<td>( dT(d1,db) )</td>
</tr>
</tbody>
</table>

\[
gives
\begin{align*}
d1 & \quad [dT(d1,ir) \land dT(d1,db)] \quad 0.9 \cdot 0.5 = 0.45 \\
d2 & \quad [l(d2,d1) \land dT(d1,ir) \land l(d2,d1) \land dT(d1,db)] \quad 0.7 \cdot 0.9 \cdot 0.5 = 0.315
\end{align*}
\]
Recursion

about(D,T) :- docTerm(D,T).
about(D,T) :- link(D,D1) & about(D1,T).

?- about(D,ir)
d1 [dT(d1,ir) | l(d1,d2) & l(d2,d3) & l(d3,d1) & dT(d1,ir) | ...] 0.900
d3 [l(d3,d1) & dT(d1,ir)] 0.720
d2 [l(d2,d3) & l(d3,d1) & dT(d1,ir)] 0.288

?- about(D,ir) & about(D,db)
d1 [dT(d1,ir) & dT(d1,db)] 0.450
d3 [l(d3,d1) & dT(d1,ir) & l(d3,d1) & dT(d1,db)] 0.360
Computation of probabilities for event expressions

1. transformation of expression into disjunctive normal form
2. application of sieve formula:
   - simple case of 2 conjuncts: \( P(a \lor b) = P(a) + P(b) - P(a \land b) \)
   - general case:
     \( c_i \) – conjunct of event keys
     \[
     P(c_1 \lor \ldots \lor c_n) = \sum_{i=1}^{n} (-1)^{i-1} \sum_{1 \leq j_1 < \ldots < j_i \leq n} P(c_{j_1} \land \ldots \land c_{j_i}).
     \]
   - \( \Rightarrow \) exponential complexity
   - \( \Rightarrow \) use only when necessary for correctness
   - see [Dalvi & Suciu 07]
Computation of probabilities for event expressions

1. transformation of expression into disjunctive normal form
2. application of sieve formula:
   - simple case of 2 conjuncts: \( P(a \lor b) = P(a) + P(b) - P(a \land b) \)
   - general case:
     \[ P(c_1 \lor \ldots \lor c_n) = \sum_{i=1}^{n} \sum_{1 \leq j_1 < \ldots < j_i \leq n} (-1)^{i-1} P(c_{j_1} \land \ldots \land c_{j_i}) \]

* \( \sim \) exponential complexity
* \( \sim \) use only when necessary for correctness
* see [Dalvi & Suciu 07]
Computation of probabilities for event expressions

1. transformation of expression into disjunctive normal form
2. application of sieve formula:
   - simple case of 2 conjuncts: \( P(a \lor b) = P(a) + P(b) - P(a \land b) \)
   - general case:
     \( c_i \) – conjunct of event keys

\[
P(c_1 \lor \ldots \lor c_n) = 
\sum_{i=1}^{n} (-1)^{i-1} \sum_{1 \leq j_1 < \ldots < j_i \leq n} P(c_{j_1} \land \ldots \land c_{j_i}).
\]

- \( \sim \) exponential complexity
- \( \sim \) use only when necessary for correctness

see [Dalvi & Suciu 07]
Computation of probabilities for event expressions

1. transformation of expression into disjunctive normal form
2. application of sieve formula:
   - simple case of 2 conjuncts: \( P(a \lor b) = P(a) + P(b) - P(a \land b) \)
   - general case:
     \( c_i \) – conjunct of event keys

\[
P(c_1 \lor \ldots \lor c_n) = \sum_{i=1}^{n} (-1)^{i-1} \sum_{1 \leq j_1 < \ldots < j_i \leq n} P(c_{j_1} \land \ldots \land c_{j_i}).
\]

- \( \sim \) exponential complexity
- \( \sim \) use only when necessary for correctness
- see [Dalvi & Suciu 07]
Possible worlds semantics

0.9 \text{docTerm}(d1,\text{ir}).

\begin{align*}
P(W_1) &= 0.9: \{\text{docTerm}(d1,\text{ir})\} \\
P(W_2) &= 0.1: \{\}
\end{align*}
IR Models based on Predicate Logic

Inference

Interpretation of probabilistic weights

0.6 docTerm(d1,ir). 0.5 docTerm(d1,db).

Possible interpretations:

\[ I_1: \ P(W_1) = 0.3: \ \{\text{docTerm}(d1,ir)\} \]
\[ P(W_2) = 0.3: \ \{\text{docTerm}(d1,ir), \ \text{docTerm}(d1,db)\} \]
\[ P(W_3) = 0.2: \ \{\text{docTerm}(d1,db)\} \]
\[ P(W_4) = 0.2: \ \{} \]

\[ I_2: \ P(W_1) = 0.5: \ \{\text{docTerm}(d1,ir)\} \]
\[ P(W_2) = 0.1: \ \{\text{docTerm}(d1,ir), \ \text{docTerm}(d1,db)\} \]
\[ P(W_3) = 0.4: \ \{\text{docTerm}(d1,db)\} \]

\[ I_3: \ P(W_1) = 0.1: \ \{\text{docTerm}(d1,ir)\} \]
\[ P(W_2) = 0.5: \ \{\text{docTerm}(d1,ir), \ \text{docTerm}(d1,db)\} \]
\[ P(W_3) = 0.4: \ \{} \]

probabilistic logic:
\[ 0.1 \leq P(\text{docTerm}(d1,ir) \& \text{docTerm}(d1,db)) \leq 0.5 \]
probabilistic Datalog with independence assumptions:
\[ P(\text{docTerm}(d1,ir) \& \text{docTerm}(d1,db)) = 0.3 \]
IR Models based on Predicate Logic

Inference

Interpretation of probabilistic weights

0.6 docTerm(d1,ir). 0.5 docTerm(d1,db).

Possible interpretations:

\( I_1: P(W_1) = 0.3: \{ \text{docTerm(d1,ir)} \} \)
\( P(W_2) = 0.3: \{ \text{docTerm(d1,ir), docTerm(d1,db)} \} \)
\( P(W_3) = 0.2: \{ \text{docTerm(d1,db)} \} \)
\( P(W_4) = 0.2: \{\} \)

\( I_2: P(W_1) = 0.5: \{ \text{docTerm(d1,ir)} \} \)
\( P(W_2) = 0.1: \{ \text{docTerm(d1,ir), docTerm(d1,db)} \} \)
\( P(W_3) = 0.4: \{ \text{docTerm(d1,db)} \} \)

\( I_3: P(W_1) = 0.1: \{ \text{docTerm(d1,ir)} \} \)
\( P(W_2) = 0.5: \{ \text{docTerm(d1,ir), docTerm(d1,db)} \} \)
\( P(W_3) = 0.4: \{\} \)

probabilistic logic:

\[ 0.1 \leq P(\text{docTerm(d1,ir)} \& \text{docTerm(d1,db)}) \leq 0.5 \]

probabilistic Datalog with independence assumptions:

\[ P(\text{docTerm(d1,ir)} \& \text{docTerm(d1,db)}) = 0.3 \]
IR Models based on Predicate Logic

Inference

Interpretation of probabilistic weights

0.6 docTerm(d1,ir). 0.5 docTerm(d1,db).

Possible interpretations:

\( I_1: P(W_1) = 0.3: \{\text{docTerm(d1,ir)}\} \)
\( P(W_2) = 0.3: \{\text{docTerm(d1,ir)}, \text{docTerm(d1,db)}\} \)
\( P(W_3) = 0.2: \{\text{docTerm(d1,db)}\} \)
\( P(W_4) = 0.2: \{\} \)

\( I_2: P(W_1) = 0.5: \{\text{docTerm(d1,ir)}\} \)
\( P(W_2) = 0.1: \{\text{docTerm(d1,ir)}, \text{docTerm(d1,db)}\} \)
\( P(W_3) = 0.4: \{\text{docTerm(d1,db)}\} \)

\( I_3: P(W_1) = 0.1: \{\text{docTerm(d1,ir)}\} \)
\( P(W_2) = 0.5: \{\text{docTerm(d1,ir)}, \text{docTerm(d1,db)}\} \)
\( P(W_3) = 0.4: \{\} \)

probabilistic logic:
0.1 \leq P(\text{docTerm(d1,ir)} \& \text{docTerm(d1,db)}) \leq 0.5

probabilistic Datalog with independence assumptions:
\( P(\text{docTerm(d1,ir)} \& \text{docTerm(d1,db)}) = 0.3 \)
Disjoint events

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>Paris</td>
<td>France</td>
</tr>
<tr>
<td>0.2</td>
<td>Paris</td>
<td>Texas</td>
</tr>
<tr>
<td>0.1</td>
<td>Paris</td>
<td>Idaho</td>
</tr>
</tbody>
</table>

Interpretation:

$P(W_1) = 0.7$: \{\text{cityState(paris, france)}\}
$P(W_2) = 0.2$: \{\text{cityState(paris, texas)}\}
$P(W_3) = 0.1$: \{\text{cityState(paris, idaho)}\}
Disjoint events

<table>
<thead>
<tr>
<th>β</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>Paris</td>
<td>France</td>
</tr>
<tr>
<td>0.2</td>
<td>Paris</td>
<td>Texas</td>
</tr>
<tr>
<td>0.1</td>
<td>Paris</td>
<td>Idaho</td>
</tr>
</tbody>
</table>

Interpretation:

\[ P(W_1) = 0.7: \{\text{cityState}(\text{paris, france})\} \]
\[ P(W_2) = 0.2: \{\text{cityState}(\text{paris, texas})\} \]
\[ P(W_3) = 0.1: \{\text{cityState}(\text{paris, idaho})\} \]
[Roelleke et al. 07]

Role of the relational Bayes: Generation of a probabilistic database
Relational Bayes
Example: \( P(\text{Nationality} \mid \text{City}) \)

<table>
<thead>
<tr>
<th>nationality_and_city</th>
</tr>
</thead>
</table>
| **Nationality** | **City**  
| "British" | "London"  
| "British" | "London"  
| "British" | "London"  
| "Scottish" | "London"  
| "French" | "London"  
| "German" | "Hamburg"  
| "German" | "Hamburg"  
| "Danish" | "Hamburg"  
| "British" | "Hamburg"  
| "German" | "Dortmund"  
| "German" | "Dortmund"  
| "Turkish" | "Dortmund"  
| "Scottish" | "Glasgow"  

<table>
<thead>
<tr>
<th>nationality_city</th>
</tr>
</thead>
</table>
| **P(Nationality|City)** | **Nationality** | **City**  
| 0.600 | "British" | "London"  
| 0.200 | "Scottish" | "London"  
| 0.200 | "French" | "London"  
| 0.500 | "German" | "Hamburg"  
| 0.250 | "Danish" | "Hamburg"  
| 0.250 | "British" | "Hamburg"  
| 0.667 | "German" | "Dortmund"  
| 0.333 | "Turkish" | "Dortmund"  
| 1.000 | "Scottish" | "Glasgow"  

1

\# \( P(\text{Nationality} \mid \text{City}) \):
2  
nationality_city \( \text{SUM}(\text{Nat}, \text{City}) :- \)
3  
nationality_and_city (Nat, City) | (City);
Relational Bayes

Example: $P(t|d)$

<table>
<thead>
<tr>
<th>Term</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>sailing</td>
<td>doc1</td>
</tr>
<tr>
<td>boats</td>
<td>doc1</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>boats</td>
<td>doc2</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>east</td>
<td>doc3</td>
</tr>
<tr>
<td>coast</td>
<td>doc3</td>
</tr>
<tr>
<td>sailing</td>
<td>doc3</td>
</tr>
<tr>
<td>sailing</td>
<td>doc4</td>
</tr>
<tr>
<td>boats</td>
<td>doc5</td>
</tr>
</tbody>
</table>

$p_{t \_d \_space}(\text{Term, DocId}) :-$

| $P(t|d)$ | Term  | DocId |
|----------|-------|-------|
| 0.50     | sailing | doc1  |
| 0.50     | boats   | doc1  |
| 0.33     | sailing | doc2  |
| 0.33     | boats   | doc2  |
| 0.33     | sailing | doc2  |
| 0.33     | east    | doc3  |
| 0.33     | coast   | doc3  |
| 0.33     | sailing | doc3  |
| 1.00     | sailing | doc4  |
| 1.00     | boats   | doc5  |

$p_{t \_d \_SUM}(\text{Term, DocId}) :-$

| $P(t|d)$ | Term  | DocId |
|----------|-------|-------|
| 0.50     | sailing | doc1  |
| 0.50     | boats   | doc1  |
| 0.67     | sailing | doc2  |
| 0.33     | boats   | doc2  |
| 0.33     | east    | doc3  |
| 0.33     | coast   | doc3  |
| 0.33     | sailing | doc3  |
| 1.00     | sailing | doc4  |
| 1.00     | boats   | doc5  |
Probabilistic rules
Rules for deterministic facts:

0.7 \text{likes-sports}(X) :- \text{man}(X).
0.4 \text{likes-sports}(X) :- \text{woman}(X).
\text{man}(peter).

\textbf{Interpretation:}
\begin{align*}
P(W_1) &= 0.7: \{\text{man}(peter), \text{likes-sports}(peter)\} \\
P(W_2) &= 0.3: \{\text{man}(peter)\} 
\end{align*}
Probabilistic rules
Rules for deterministic facts:

0.7 likes-sports(X) :- man(X).
0.4 likes-sports(X) :- woman(X).
man(peter).

Interpretation:

\[ P(W_1) = 0.7: \{ \text{man(peter)}, \text{likes-sports(peter)} \} \]
\[ P(W_2) = 0.3: \{ \text{man(peter)} \} \]
Probabilistic rules
Rules for uncertain facts:

# gender is disjoint on the first attribute
0.7 \( l\text{-sports}(X) \) :- gender(X,male).
0.4 \( l\text{-sports}(X) \) :- gender(X,female).
0.5 gender(X,male) :- human(X).
0.5 gender(X,female) :- human(X).
human(jo).

Interpretation:
\[ P(W_1) = 0.35: \{ \text{gender(jo,male), l-sports(jo)} \} \]
\[ P(W_2) = 0.15: \{ \text{gender(jo,male)} \} \]
\[ P(W_3) = 0.20: \{ \text{gender(jo,female), l-sports(jo)} \} \]
\[ P(W_4) = 0.30: \{ \text{gender(jo,female)} \} \]

?- l-sports(jo) \[ P(W_1) + P(W_3) = 0.55 \]
Probabilistic rules
Rules for uncertain facts:

\# gender is disjoint on the first attribute
0.7 \text{ l-sports}(X) :- \text{ gender}(X,\text{male}).
0.4 \text{ l-sports}(X) :- \text{ gender}(X,\text{female}).
0.5 \text{ gender}(X,\text{male}) :- \text{ human}(X).
0.5 \text{ gender}(X,\text{female}) :- \text{ human}(X).
\text{human}(jo).

\textbf{Interpretation:}
\[ P(W_1) = 0.35: \{\text{gender}(jo,\text{male}), \text{ l-sports}(jo)\} \]
\[ P(W_2) = 0.15: \{\text{gender}(jo,\text{male})\} \]
\[ P(W_3) = 0.20: \{\text{gender}(jo,\text{female}), \text{ l-sports}(jo)\} \]
\[ P(W_4) = 0.30: \{\text{gender}(jo,\text{female})\} \]

?- \text{ l-sports}(jo) \quad P(W_1) + P(W_3) = 0.55
# gender is disjoint on the first attribute
0.7 l-sports(X) :- gender(X,male).
0.4 l-sports(X) :- gender(X,female).
0.5 gender(X,male) :- human(X).
0.5 gender(X,female) :- human(X).
human(jo).

**Interpretation:**

\[ P(W_1) = 0.35: \{ \text{gender(jo,male), l-sports(jo)} \} \]
\[ P(W_2) = 0.15: \{ \text{gender(jo,male)} \} \]
\[ P(W_3) = 0.20: \{ \text{gender(jo,female), l-sports(jo)} \} \]
\[ P(W_4) = 0.30: \{ \text{gender(jo,female)} \} \]

\[ P(W_1) + P(W_3) = 0.55 \]
Probabilistic rules
Rules for independent events

\[
\text{sameauthor}(D_1, D_2) :\neg \text{author}(D_1, X) \land \text{author}(D_2, X).
\]

\[
0.5 \text{ link}(D_1, D_2) :\neg \text{refer}(D_1, D_2).
\]

\[
0.2 \text{ link}(D_1, D_2) :\neg \text{sameauthor}(D_1, D_2).
\]

\[
?? \text{ link}(D_1, D_2) :\neg \text{refer}(D_1, D_2) \land \text{sameauthor}(D_1, D_2).
\]

\[
P(l\mid r), P(l\mid s) \rightarrow P(l\mid r \land s)?
\]
Rules for independent events
Modeling probabilistic inference networks

0.7 link(D1,D2) :- refer(D1,D2) & sameauthor(D1,D2).
0.5 link(D1,D2) :- refer(D1,D2) & not(sameauthor(D1,D2)).
0.2 link(D1,D2) :- sameauthor(D1,D2) & not(refer(D1,D2)).

Probabilistic inference networks,
rules define link matrix
Vague Predicates

- Disjoint events
- Relational Bayes
- Probabilistic rules
- The Logical View on Vague Predicates
- Vague Predicates in IR and Databases
- Probabilistic Modeling of Vague Predicates
Vague Predicates
Motivating Example

"lcd tv 46inch"

Showing 1 - 16 of 3,851 Results

**Samsung LN46E550 46-Inch 1080p 60Hz LCD HDTV** by Samsung

$879.99 Click for product details
Order in the next **5 hours** and get it by **Wednesday, Jan 16**.

More Buying Choices
$**463.80** used & new (14 offers)

**Samsung LN46D550 46-Inch 1080p 60Hz LCD HDTV (Black)** by Samsung

$899.99 $**599.27**
Only 15 left in stock - order soon.

More Buying Choices
$**599.27** new (4 offers)
$**490.00** used (10 offers)

**Cheetah Mounts APTMM2B Flush Tilt Dual Hook (1.3" from wall) Flat Screen Cheetah**

$49.99 $**27.99**
Order in the next **7 hours** and get it by **Wednesday, Jan 16**.

More Buying Choices
$**27.99** new (9 offers)
Vague Predicates
Motivating Example

"lcd tv 45inch"

Showing 1 - 16 of 2,617 Results

RCA 32LB45RQ 32-Inch Full 1080p 60Hz LCD HDTV by RCA
$228.38 used (4 offers)

RCA 42LB45RQ 42-Inch 1080p 60Hz LCD HDTV (Black) by RCA
$476.99
Only 1 left in stock - order soon.

RCA 22LB45RQD 22-Inch Full 1080p LCD/DVD Combo HDTV by RCA
$229.99 $219.99
Only 1 left in stock - order soon.

More Buying Choices
$219.99 new (2 offers)
$188.99 new (3 offers)
$125.00 used (19 offers)
Current IR systems are based on proposition logic (query term present/absent in document)

- Similarity of values not considered
- but multimedia IR deals with similarity already
- transition from propositional to predicate logic necessary
Propositional vs. Predicate Logic

- Current IR systems are based on proposition logic (query term present/absent in document)
- Similarity of values not considered
  - but multimedia IR deals with similarity already
  - \(\sim\) transition from propositional to predicate logic necessary
Propositional vs. Predicate Logic

- Current IR systems are based on proposition logic (query term present/absent in document)
- Similarity of values not considered
- but multimedia IR deals with similarity already
- ~ transition from propositional to predicate logic necessary
Propositional vs. Predicate Logic

- Current IR systems are based on proposition logic (query term present/absent in document)
- Similarity of values not considered
- but multimedia IR deals with similarity already
  - transition from propositional to predicate logic necessary
Propositional vs. Predicate Logic

- Current IR systems are based on proposition logic (query term present/absent in document)
- Similarity of values not considered
- but multimedia IR deals with similarity already
- \( \leadsto \) transition from propositional to predicate logic necessary
Vague Predicates in Probabilistic Datalog

[Fuhr & Roelleke 97] [Fuhr 00]

- Example: Shopping 45 inch LCD TV
- Vague predicates as builtin predicates:
  \( X \approx Y \)
- query(D):- Category(D, tv) & type(D, lcd) & size(D, X) & \( \approx (X, 45) \)

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>42</td>
<td>45</td>
</tr>
<tr>
<td>0.8</td>
<td>43</td>
<td>45</td>
</tr>
<tr>
<td>0.9</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>1.0</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>0.9</td>
<td>46</td>
<td>45</td>
</tr>
<tr>
<td>0.8</td>
<td>47</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Data types and vague predicates in IR

Data type: domain + (vague) predicates

- Language (multilingual documents) / (language-specific stemming)
- Person names / “his name sounds like Jones”
- Dates / “about a month ago”
- Amounts / “orders exceeding 1 Mio $”
- Technical measurements / “at room temperature”
- Chemical formulas
Vague Criteria in Fact Databases

"I am looking for a 45-inch LCD TV with
- wide viewing angle
- high contrast
- low price
- high user rating"

→ vague criteria are very frequent in end-user querying of fact databases
→ but no appropriate support in SQL

Vague conditions → similar to fuzzy predicates
Vague Criteria in Fact Databases

"I am looking for a 45-inch LCD TV with
- wide viewing angle
- high contrast
- low price
- high user rating"

→ vague criteria are very frequent in end-user querying of fact databases
→ but no appropriate support in SQL

vague conditions → similar to fuzzy predicates
Probabilistic Modeling of Vague Predicates

[Fuhr 90]

- learn vague predicates from feedback data
- construct feature vector $\vec{x}(q_i, d_i)$ from query value $q_i$ and document value $d_i$ (e.g. relative difference)
- apply logistic regression
Expressiveness

- Disjoint events
- Relational Bayes
- Probabilistic rules
- Retrieval Rules, Joins, Aggregations and Restructuring
- Expressiveness in XML Retrieval
Expressiveness
Formulating Retrieval Rules

about(D,T) :- docTerm(D,T).

consider document linking / anchor text
about(D,T) :- link(D1,D), about(D1,T).

consider term hierarchy
about(D,T) :- subconcept(T,T1) & about(D,T1).

field-specific term weighting
0.9 docTerm(D,T) :- occurs(D,T,title).
0.5 docTerm(D,T) :- occurs(D,T,`body`).
Expressiveness
Formulating Retrieval Rules

\[\text{about}(D,T) \leftarrow \text{docTerm}(D,T).\]

consider document linking / anchor text
\[\text{about}(D,T) \leftarrow \text{link}(D1,D),\text{about}(D1,T).\]

consider term hierarchy
\[\text{about}(D,T) \leftarrow \text{subconcept}(T,T1) \& \text{about}(D,T1).\]

field-specific term weighting
\[0.9 \ \text{docTerm}(D,T) \leftarrow \text{occurs}(D,T,\text{title}).\]
\[0.5 \ \text{docTerm}(D,T) \leftarrow \text{occurs}(D,T,\text{body}).\]
Expressiveness
Formulating Retrieval Rules

\begin{align*}
\text{about}(D,T) & : \text{docTerm}(D,T). \\
\text{consider document linking / anchor text} & \\
\text{about}(D,T) & : \text{link}(D1,D), \text{about}(D1,T). \\
\text{consider term hierarchy} & \\
\text{about}(D,T) & : \text{subconcept}(T,T1) \land \text{about}(D,T1). \\
\text{field-specific term weighting} & \\
0.9 \text{ docTerm}(D,T) & : \text{occurs}(D,T,\text{title}). \\
0.5 \text{ docTerm}(D,T) & : \text{occurs}(D,T,\text{body}).
\end{align*}
Expressiveness
Formulating Retrieval Rules

about(D,T) :- docTerm(D,T).

consider document linking / anchor text
about(D,T) :- link(D1,D), about(D1,T).

consider term hierarchy
about(D,T) :- subconcept(T,T1) & about(D,T1).

field-specific term weighting
0.9 docTerm(D,T) :- occurs(D,T,title).
0.5 docTerm(D,T) :- occurs(D,T,body).
**Expressiveness**

**Joins**

*IR authors:*

\[
\text{irauthor}(N) :\neg \ \text{about}(D, \text{ir}) \ \& \ \text{author}(D,N).
\]

*Smith's IR papers cited by Miller*

\[
? \neg \ \text{author}(D,\text{smith}) \ \& \ \text{about}(D, \text{ir}) \ \& \\
\text{author}(D1,\text{miller}) \ \& \ \text{cites}(D,D1).
\]
Expressiveness

Joins

IR authors:

irauthor(N):- about(D,ir) & author(D,N).

Smith’s IR papers cited by Miller

?- author(D,smith) & about(D,ir) &
    author(D1,miller) & cites(D,D1).
### Expressiveness

#### Aggregation (1)

Who are the major IR authors?

<table>
<thead>
<tr>
<th>β</th>
<th>DNO</th>
<th>TERM</th>
<th>author</th>
<th>DNO</th>
<th>NAME</th>
<th>irauthor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>1</td>
<td>ir</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>db</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>ir</td>
<td></td>
<td>1</td>
<td>smith</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>ir</td>
<td></td>
<td>2</td>
<td>miller</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>3</td>
<td>ai</td>
<td></td>
<td>3</td>
<td>smith</td>
<td>0.98</td>
</tr>
</tbody>
</table>

\[
\text{irauthor}(A) :\!-\! \text{index}(D, \text{ir}) \land \text{author}(D, A).
\]

Aggregation through projection!
Who are the major IR authors?

<table>
<thead>
<tr>
<th>β</th>
<th>DNO</th>
<th>TERM</th>
<th>author</th>
<th>DNO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>1</td>
<td>ir</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>db</td>
<td></td>
<td>1</td>
<td>smith</td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>ir</td>
<td></td>
<td>2</td>
<td>miller</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>ir</td>
<td></td>
<td>3</td>
<td>smith</td>
</tr>
<tr>
<td>0.7</td>
<td>3</td>
<td>ai</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

irauthor(A):- index(D,ir) & author(D,A).

Aggregation through projection!
Who are the major IR authors?

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>DNO</th>
<th>TERM</th>
<th>author</th>
<th>irauths</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>1</td>
<td>ir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>db</td>
<td>1</td>
<td>smith</td>
</tr>
<tr>
<td>0.6</td>
<td>2</td>
<td>ir</td>
<td>2</td>
<td>miller</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>ir</td>
<td>3</td>
<td>smith</td>
</tr>
<tr>
<td>0.7</td>
<td>3</td>
<td>ai</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Aggregation through summing:

\[ \text{irauth}(D,A) :- \text{index}(D,ir) \land \text{author}(D,A). \]

\[ \text{irauths} \text{ SUM}(\text{Name}) :- \text{irdbauth}(\text{Doc},\text{Name}) \lor (\text{Name}) \]
[Fuhr & Lalmas 07]

Content Typing

Object Types

Data Types

Text only

Structure

Nested structure

Named fields

XPath

XQuery
Expressiveness in XML Retrieval

[Fuhr & Lalmas 07]

Content Typing

Object Types

Data Types

Text only

Structure

Nested structure
Named fields
XPath
XQuery
XML structure: 1. Nested Structure

- XML document as hierarchical structure
- Retrieval of elements (subtrees)
- Typical query language does not allow for specification of structural constraints
- Relevance-oriented selection of answer elements: return the most specific relevant elements
XML structure: 2. Named Fields

- Reference to elements through field names only
- Context of elements is ignored (e.g. author of article vs. author of referenced paper)
- Post-Coordination may lead to false hits (e.g. author name – author affiliation)

Example: Dublin Core

```xml
  <dc:title>Generic Algebras</dc:title>
  <dc:creator>A. Smith (ESI), B. Miller (CMU)</dc:creator>
  <dc:subject>Orthogonal group, Symplectic group</dc:subject>
  <dc:date>2001-02-27</dc:date>
  <dc:format>application/postscript</dc:format>
  <dc:source>ESI preprints</dc:source>
  <dc:language>en</dc:language>
</oai_dc:dc>
```
Expressiveness in XML Retrieval

XML structure: 3. XPath

```
/document/chapter[about(./heading, XML) AND
about(./section//*,syntax)]
```
XPath expression:

```
/document/chapter[about(./heading, XML) AND about(./section//*, syntax)]
```
XML structure: 3. XPath (cont’d)

- Full expressiveness for navigation through document tree (+links)
  - Parent/child, ancestor/descendant
  - Following/preceding, following-sibling, preceding-sibling
  - Attribute, namespace
- Selection of arbitrary elements/subtrees
  (but answer can be only a single element of the originating document)
XML structure: 4. XQuery

Higher expressiveness, especially for database-like applications:

- Joins (trees $\rightarrow$ graphs)
- Aggregations
- Constructors for restructuring results

Example: List each publisher and the average price of its books

FOR $p$ IN distinct(document("bib.xml")//publisher)
LET $a := \text{avg}(\text{document("bib.xml")//book[publisher = $p]}/\text{price})$
RETURN
<publisher>
  <name> $p$/text() </name>
  <avgprice> $a$ </avgprice>
</publisher>
Higher expressiveness, especially for database-like applications:

- Joins (trees $\rightarrow$ graphs)
- Aggregations
- Constructors for restructuring results

Example: List each publisher and the average price of its books

```xquery
FOR $p$ IN distinct(document("bib.xml")//publisher)
LET $a$ := avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
  <publisher>
    <name> $p/text() </name>
    <avgprice> $a </avgprice>
  </publisher>
```
XML content typing

Content Typing

Object Types

Data Types

Text only

Nested structure

Named fields

XPath

XQuery

Structure
XML content typing: 1. Text

<book>
  <author>John Smith</author>
  <title>XML Retrieval</title>
  <chapter> <heading>Introduction</heading>
    This text explains all about XML and IR.
  </chapter>
  <chapter>
    <heading> XML Query Language XQL </heading>
    <section>
      <heading>Examples</heading>
    </section>
    <section>
      <heading>Syntax</heading>
      Now we describe the XQL syntax.
    </section>
  </chapter>
</book>

Example query

//chapter[about(., XML query language]
XML content typing: 2. Data Types

- Data type: domain + (vague) predicates (see above)
- Close relationship to XML Schema, but
  - XMLS supports syntactic type checking only
  - No support for vague predicates
Object types: Persons, Locations, Dates, ....

*Pablo Picasso* (October 25, 1881 - April 8, 1973) was a Spanish painter and sculptor. In *Paris*, Picasso entertained a distinguished coterie of friends in the Montmartre and Montparnasse quarters, including André Breton, Guillaume Apollinaire, and writer Gertrude Stein.

To which other artists did Picasso have close relationships? Did he ever visit the USA?

Named entity recognition methods allow for automatic markup of object types

Object types support increased precision
XML content typing
Tag semantics modelled as hierarchies

Object type hierarchies

- Person
  - Scientist
    - Physicist
  - Artist
    - Poet
    - Actor
    - Singer

Role hierarchies

- Creator
  - Author
  - Editor
XML content typing
Tag semantics modelled in OWL
Description Logic/Ontologies

- Disjoint events
- Relational Bayes
- Probabilistic rules
- Thesaurus
- Introduction into OWL
- SPARQL
thesaurus knowledge:
can be expressed in propositional logic
\[ \text{square} = \text{quadrangle} \land \text{regular-polygon} \]

description logic
- based on semantic networks
- more expressive than thesauri
- instances of concepts
- roles between (instances of) concepts
thesaurus knowledge:
can be expressed in propositional logic
square = quadrangle \land \text{regular-polygon}

description logic
- based on semantic networks
- more expressive than thesauri
  - instances of concepts
  - roles between (instances of) concepts
IR Models based on Predicate Logic

Description Logic/Ontologies

Thesaurus

thesaurus knowledge:
can be expressed in propositional logic
square = quadrangle \land \text{regular-polygon}

description logic
- based on semantic networks
- more expressive than thesauri
  - instances of concepts
  - roles between (instances of) concepts
thesaurus knowledge:
can be expressed in propositional logic
\( \text{square} = \text{quadrangle} \land \text{regular-polygon} \)

description logic
- based on semantic networks
- more expressive than thesauri
  - instances of concepts
  - roles between (instances of) concepts
IR Models based on Predicate Logic

Description Logic/Ontologies

Thesaurus

thesaurus knowledge:
can be expressed in propositional logic
square = quadrangle \land \text{regular-polygon}

description logic
- based on semantic networks
- more expressive than thesauri
  - instances of concepts
  - roles between (instances of) concepts
Semantic Web (ontology) languages

RDF: “Resource description language”
semantics markup language, only resources and their properties, serialisation in XML

RDFS: “RDF Schema”, schema definition language for RDF

OWL: extends RDF/RDFS by richer modelling primitives,
OWL Lite/DL/Full
  - OWL Lite contains simple primitives
  - OWL DL corresponds to expressive description logic
  - OWL Full is OWL DL + RDF

knowledge base can be modelled as collection of RDF triples (RDF/XML serialisation)
alternative encoding: abstract syntax (easier to read)
Semantic Web (ontology) languages

RDF: “Resource description language”
semantic markup language, only resources and their
properties, serialisation in XML

RDFS: “RDF Schema”, schema definition language for RDF

OWL: extends RDF/RDFS by richer modelling primitives,
OWL Lite/DL/Full
  - OWL Lite contains simple primitives
  - OWL DL corresponds to expressive description
    logic
  - OWL Full is OWL DL + RDF

knowledge base can be modelled as collection of RDF
triples (RDF/XML serialisation)
alternative encoding: abstract syntax (easier to read)
Semantic Web (ontology) languages

**RDF:** “Resource description language”
- semantic markup language, only resources and their properties, serialisation in XML

**RDFS:** “RDF Schema”, schema definition language for RDF

**OWL:** extends RDF/RDFS by richer modelling primitives,
- OWL Lite/DL/Full
  - OWL Lite contains simple primitives
  - OWL DL corresponds to expressive description logic
  - OWL Full is OWL DL + RDF
- knowledge base can be modelled as collection of RDF triples (RDF/XML serialisation)
- alternative encoding: abstract syntax (easier to read)
Semantic Web (ontology) languages

**RDF**: “Resource description language”
semantic markup language, only resources and their properties, serialisation in XML

**RDFS**: “RDF Schema”, schema definition language for RDF

**OWL**: extends RDF/RDFS by richer modelling primitives, OWL Lite/DL/Full
  
  - **OWL Lite** contains simple primitives
  - **OWL DL** corresponds to expressive description logic
  - **OWL Full** is OWL DL + RDF

knowledge base can be modelled as collection of RDF triples (RDF/XML serialisation)
alternative encoding: abstract syntax (easier to read)
Semantic Web (ontology) languages

**RDF:** “Resource description language”
semantic markup language, only resources and their properties, serialisation in XML

**RDFS:** “RDF Schema”, schema definition language for RDF

**OWL:** extends RDF/RDFS by richer modelling primitives, OWL Lite/DL/Full
- OWL Lite contains simple primitives
- OWL DL corresponds to expressive description logic
- OWL Full is OWL DL + RDF

knowledge base can be modelled as collection of RDF triples (RDF/XML serialisation)
alternative encoding: abstract syntax (easier to read)
Semantic Web (ontology) languages

**RDF:** “Resource description language”
semantic markup language, only resources and their properties, serialisation in XML

**RDFS:** “RDF Schema”, schema definition language for RDF

**OWL:** extends RDF/RDFS by richer modelling primitives, OWL Lite/DL/Full
- OWL Lite contains simple primitives
- OWL DL corresponds to expressive description logic
- OWL Full is OWL DL + RDF

knowledge base can be modelled as collection of RDF triples (RDF/XML serialisation)
alternative encoding: abstract syntax (easier to read)
Objects, classes, literals and datatypes

- **Two distinct domains:**
  - **Classes:** for objects
  - **Data types:** for literals

![Diagram showing the relationship between classes and data types in OWL]

- `owl:Class`
  - `rdf:type` to `Person`
  - `Person` is a subclass of `owl:Class`

- `owl:Datatype`
  - `rdf:type` to `xsd:decimal`
  - `xsd:decimal` is a data type for literals

- `Person` is an instance of `owl:Class` and has type `rdf:type`

- `1,89` is an instance of `xsd:decimal` and has type `rdf:type`
Class(Female partial Animal)

<owl:Class rdf:ID="Female">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
Class(Male partial Animal)
DisjointClasses(Male Female)

<owl:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <owl:disjointWith rdf:resource="#Female"/>
</owl:Class>
Object properties (1)

ObjectProperty(hasParent domain(Animal) range(Animal))

<owl:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</owl:Class>
Object properties (2)

```
ObjectProperty(hasFather super(hasParent) range(Male))

<owl:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</owl:Class>
```
Datatype properties

DatatypeProperty(shoesize Functional domain(Animal) range(xsd:decimal))

<owl:DatatypeProperty rdf:ID="shoesize">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="xsd:decimal"/>
  <rdf:type rdf:resource="owl:FunctionalProperty"/>
</owl:Class>
Property restrictions

Class(Person partial Animal restriction(hasParent allValuesFrom(Person)
    restriction(hasParent cardinality(2)))

<owl:Class rdf:ID="Person">
    <rdfs:subClassOf rdf:resource="#Animal"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasParent"/>
            <owl:allValuesFrom rdf:resource="#Person"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasParent"/>
            <owl:cardinality>2</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
Instances

Individual(Kain type(Male) value(hasFather Adam) value(hasMother Eve) value(shoeseize 10))

<Male rdf:ID="Kain">  
  <hasFather rdf:resource="#Adam"/>  
  <hasMother rdf:resource="#Eve"/>  
  <shoeseize>10</shoeseize>  
</Male>
Further modelling primitives

`owl:inverseOf`: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

`owl:TransitiveProperty`: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

`owl:SymmetricProperty`: \( p(a, b) \rightarrow p(b, a) \)

`owl:InverseFunctionalProperty`: inverse property is functional

`owl:hasValue` at least one property value equals object or datatype value

`owl:someValuesFrom` at least one property value is instance of class, expression or datatype

`owl:interSectionOf`, `owl:unionOf`, `owl:complementOf`: boolean combinations of class expressions

`owl:oneOf`: define class by enumerating its instances
Further modelling primitives

- **owl:inverseOf**: inverse property: $p(a, b) \leftrightarrow r(b, a)$
- **owl:TransitiveProperty**: $p(a, b), p(b, c) \rightarrow p(a, c)$
- **owl:SymmetricProperty**: $p(a, b) \rightarrow p(b, a)$
- **owl:InverseFunctionalProperty**: inverse property is functional
- **owl:hasValue**: at least one property value equals object or datatype value
- **owl:someValuesFrom**: at least one property value is instance of class, expression or datatype
- **owl:intersectionOf, owl:unionOf, owl:complementOf**: boolean combinations of class expressions
- **owl:oneOf**: define class by enumerating its instances
Further modelling primitives

**owl:inverseOf:** inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

**owl:TransitiveProperty:** \( p(a, b), p(b, c) \rightarrow p(a, c) \)

**owl:SymmetricProperty:** \( p(a, b) \rightarrow p(b, a) \)

**owl:InverseFunctionalProperty:** inverse property is functional

**owl:hasValue** at least one property value equals object or datatype value

**owl:someValuesFrom** at least one property value is instance of class, expression or datatype

**owl:interSectionOf, owl:unionOf, owl:complementOf**: boolean combinations of class expressions

**owl:oneOf**: define class by enumerating its instances
Further modelling primitives

owl:inverseOf: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

owl:TransitiveProperty: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

owl:SymmetricProperty: \( p(a, b) \rightarrow p(b, a) \)

owl:InverseFunctionalProperty: inverse property is functional

owl:hasValue: at least one property value equals object or datatype value

owl:someValuesFrom: at least one property value is instance of class, expression or datatype

owl:intersectionOf, owl:unionOf, owl:complementOf: boolean combinations of class expressions

owl:oneOf: define class by enumerating its instances
Further modelling primitives

owl:inverseOf: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

owl:TransitiveProperty: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

owl:SymmetricProperty: \( p(a, b) \rightarrow p(b, a) \)

owl:InverseFunctionalProperty: inverse property is functional

owl:hasValue at least one property value equals object or datatype value

owl:someValuesFrom at least one property value is instance of class, expression or datatype

owl:intersectionOf, owl:unionOf, owl:complementOf: boolean combinations of class expressions

owl:oneOf: define class by enumerating its instances
Further modelling primitives

**owl:inverseOf**: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

**owl:TransitiveProperty**: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

**owl:SymmetricProperty**: \( p(a, b) \rightarrow p(b, a) \)

**owl:InverseFunctionalProperty**: inverse property is functional

**owl:hasValue** at least one property value equals object or datatype value

**owl:someValuesFrom** at least one property value is instance of class, expression or datatype

**owl:interSectionOf, owl:unionOf, owl:complementOf**: boolean combinations of class expressions

**owl:oneOf**: define class by enumerating its instances
Further modelling primitives

owl:inverseOf: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

owl:TransitiveProperty: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

owl:SymmetricProperty: \( p(a, b) \rightarrow p(b, a) \)

owl:InverseFunctionalProperty: inverse property is functional

owl:hasValue at least one property value equals object or datatype value

owl:someValuesFrom at least one property value is instance of class, expression or datatype

owl:intersectionOf, owl:unionOf, owl:complementOf: boolean combinations of class expressions

owl:oneOf: define class by enumerating its instances
Further modelling primitives

`owl:inverseOf`: inverse property: \( p(a, b) \leftrightarrow r(b, a) \)

`owl:TransitiveProperty`: \( p(a, b), p(b, c) \rightarrow p(a, c) \)

`owl:SymmetricProperty`: \( p(a, b) \rightarrow p(b, a) \)

`owl:InverseFunctionalProperty`: inverse property is functional

`owl:hasValue` at least one property value equals object or datatype value

`owl:someValuesFrom` at least one property value is instance of class, expression or datatype

`owl:interSectionOf, owl:unionOf, owl:complementOf`: boolean combinations of class expressions

`owl:oneOf`: define class by enumerating its instances
Limitations of OWL

OWL lacks support for

1. **uncertainty:** only deterministic relationships possible, no weighting or probabilistic facts
   ⇒ “\( \text{Pr}(\text{hasFather(lisa, thomas)}) = 0.9 \)” cannot be expressed

2. **rules:** no general rules,
   only specific rules like subClassOf, TransitiveProperty ...
   ⇒ “if hasParent(A, B) and hasParent(C, D) and hasSibling(B, D), then hasCousin(A, C)” cannot be expressed

3. **n-ary datatype predicates:**
   OWL datatypes are based on XML Schema datatypes, thus providing only unary datatype predicates
   ⇒ “sameDomain(foo@bar.de, baz@bar.de)” cannot be expressed

⇒ IR queries cannot be expressed directly in OWL
Limitations of OWL

OWL lacks support for

- **uncertainty**: only deterministic relationships possible, no weighting or probabilistic facts
  ⇒ “Pr(hasFather(lisa, thomas))=0.9” cannot be expressed

- **rules**: no general rules, only specific rules like subClassOf, TransitiveProperty . . .
  ⇒ “if hasParent(A, B) and hasParent(C, D) and hasSibling(B, D), then hasCousin(A, C)” cannot be expressed

- **n-ary datatype predicates**: OWL datatypes are based on XML Schema datatypes, thus providing only unary datatype predicates
  ⇒ “sameDomain(foo@bar.de, baz@bar.de)” cannot be expressed

  ⇒ IR queries cannot be expressed directly in OWL
Limitations of OWL

OWL lacks support for

- **uncertainty**: only deterministic relationships possible, no weighting or probabilistic facts
  ⇒ “Pr(hasFather(lisa, thomas)) = 0.9” cannot be expressed

- **rules**: no general rules, only specific rules like subClassOf, TransitiveProperty ...
  ⇒ “if hasParent(A, B) and hasParent(C, D) and hasSibling(B, D), then hasCousin(A, C)” cannot be expressed

- **n-ary datatype predicates**: OWL datatypes are based on XML Schema datatypes, thus providing only unary datatype predicates
  ⇒ “sameDomain(foo@bar.de, baz@bar.de)” cannot be expressed

⇒ IR queries cannot be expressed directly in OWL
Limitations of OWL

OWL lacks support for

- **uncertainty**: only deterministic relationships possible, no weighting or probabilistic facts
  ⇒ “Pr(hasFather(lisa, thomas))=0.9” cannot be expressed

- **rules**: no general rules, only specific rules like subClassOf, TransitiveProperty...
  ⇒ “if hasParent(A,B) and hasParent(C,D) and hasSibling(B,D), then hasCousin(A,C)” cannot be expressed

- **n-ary datatype predicates**: OWL datatypes are based on XML Schema datatypes, thus providing only unary datatype predicates
  ⇒ “sameDomain(foo@bar.de, baz@bar.de)” cannot be expressed

⇒ IR queries cannot be expressed directly in OWL
Limitations of OWL

OWL lacks support for

- **uncertainty**: only deterministic relationships possible, no weighting or probabilistic facts
  $\Rightarrow$ “$\text{Pr}(\text{hasFather}(\text{lisa}, \text{thomas})) = 0.9$” cannot be expressed

- **rules**: no general rules, only specific rules like subClassOf, TransitiveProperty . . .
  $\Rightarrow$ “if hasParent(A,B) and hasParent(C,D) and hasSibling(B,D), then hasCousin(A,C)” cannot be expressed

- **n-ary datatype predicates**: OWL datatypes are based on XML Schema datatypes, thus providing only unary datatype predicates
  $\Rightarrow$ “sameDomain(foo@bar.de, baz@bar.de)” cannot be expressed

$\Rightarrow$ IR queries cannot be expressed directly in OWL
OWL: Conclusion

- OWL extends RDF(S) by additional modelling primitives
- well-defined semantics, based on description logics
- does not support all RDF features (no reification, only three levels owl:Class, classes and objects)
- lacks important features:
  - only deterministic features, no probabilistic relationships
  - no rules (but in SWRL)
  - restricted datatype predicates (due to XML Schema)
- OWL and associated languages become standard in the Semantic Web
Semantic Web Layers

- Self-desc. doc.
- Data
- Rules
- Proof
- Trust

- Ontology vocabulary
- RDF + rdfschema
- XML + NS + xmlschema
- Unicode
- URI
SPARQL

query language for getting information from RDF (OWL) graphs

Facilities for

- extract information in the form of URIs, blank nodes, plain and typed literals
- extract RDF subgraphs
- construct new RDF graphs based on information in the queried graphs

Features:

- matching graph patterns
- variables – global scope; indicated by '?' or '$'
SPARQL: Basic Graph Pattern

- **Set of Triple Patterns**
  - Triple Pattern – similar to an RDF Triple (subject, predicate, object), but any component can be a query variable; literal subjects are allowed
  - Matching a triple pattern to a graph: bindings between variables and RDF Terms

- **Matching of Basic Graph Patterns**
  - A Pattern Solution of Graph Pattern GP on graph G is any substitution S such that S(GP) is a subgraph of G.
    - `SELECT ?x ?v WHERE ?x ?v ?x`
**SPARQL: Group Patterns + Value Constraints**

Group Pattern: A set of graph patterns which must all match
Value Constraints: restrict RDF terms in a solution

```
SELECT ?n WHERE
?n profession "Physicist" . ?n isa "Politician"
```
**SPARQL: Query forms**

**SELECT** returns all, or a subset of the variables bound in a query pattern match.

**formats**: XML or RDF/XML

**CONSTRUCT** returns an RDF graph constructed by substituting variables in a set of triple templates.

**DESCRIBE** returns an RDF graph that describes the resources found.

**ASK** returns whether a query pattern matches or not.
Conclusion and Outlook

- Disjoint events
- Relational Bayes
- Probabilistic rules
Conclusion

Inference

- Probabilistic relational model supports integration of IR+DB
- Probabilistic Datalog as powerful inference mechanism
- Allows for formulating retrieval strategies as logical rules

Vague predicates

- Natural extension of IR methods to attribute values
- Vague predicates can be learned from feedback data
- Transition from propositional to predicate logic

Expressive query language

- Joins
- Aggregations
- (Re)structuring of results
Conclusion

Inference

- Probabilistic relational model supports integration of IR+DB
- Probabilistic Datalog as powerful inference mechanism
- Allows for formulating retrieval strategies as logical rules

Vague predicates

- Natural extension of IR methods to attribute values
- Vague predicates can be learned from feedback data
- Transition from propositional to predicate logic

Expressive query language

- Joins
- Aggregations
- (Re)structuring of results
Conclusion

Inference

- Probabilistic relational model supports integration of IR+DB
- Probabilistic Datalog as powerful inference mechanism
- Allows for formulating retrieval strategies as logical rules

Vague predicates

- Natural extension of IR methods to attribute values
- Vague predicates can be learned from feedback data
- Transition from propositional to predicate logic

Expressive query language

- Joins
- Aggregations
- (Re)structuring of results
The HySpirit software development kit provides a description-based approach for modelling complex information retrieval tasks such as hypermedia and knowledge retrieval, by combining database models, probability theory, logic and object-oriented concepts for the representation of knowledge and its intrinsic uncertainty.

Layered Architecture

Heterogeneous Data
Structured Documents
MPEG-7
XML
HTML
Hypertext
User Profiles
Flat File
Complex Objects
Semi-Structured Documents

Proabilistic Relational Algebra
- Database Model & Interface
- Implementation

Probabilistic Cataloging
- Incomplete, ambiguous and inconsistent knowledge representation
- Uncertainty reasoning

Probabilistic Object-Oriented Logic
- Knowledge representation leveraging
- Inheritance in databases
- Inference and query resolution

-era
University of Dortmund
Queen Mary, University of London
Hyspirit GmbH

References


http://www.eecs.qmul.ac.uk/~thor/
Don’t *program* search engines, *design* them

http://www.spinque.com/
Outlook
IR Systems vs. DBMS
Outlook

IR Systems vs. DBMS

- Application
  - DBMS
  - DB

- Pragmatics

- IRS
  - Collection
Outlook
IR Systems vs. DBMS

Separation between IRS and IR application?
Towards an IRMS
Towards an IRMS

Application

DBMS

SQL

Application

IRMSS

Collection
Towards an IRMS

Application

DBMS

DB

SQL

IR Query Language

Application

IRMS

Collection
References I

Dalvi, N. N.; Suciu, D.
(2007).
Efficient query evaluation on probabilistic databases.
VLDB J. 16(4), pages 523–544.

Forst, J. F.; Tombros, A.; Roelleke, T.
(2007).
POLIS: A Probabilistic Logic for Document Summarisation.

Frommholz, I.; Fuhr, N.
Probabilistic, Object-oriented Logics for Annotation-based Retrieval in Digital Libraries.
Fuhr, N.; Lalmas, M.
(2007).
Advances in XML retrieval: the INEX initiative.

Fuhr, N.; Rölleke, T.
(1997).
A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. 

Fuhr, N.; Rölleke, T.
HySpirit – a Probabilistic Inference Engine for Hypermedia Retrieval in Large Databases.
A Probabilistic Framework for Vague Queries and Imprecise Information in Databases. 

Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications.
*Journal of the American Society for Information Science 51*(2), pages 95–110.

Intelligent Hypermedia Retrieval. 
Pearl, J.
(1988).
*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.*
Morgan Kaufman, San Mateo, California.

Rölleke, T.; Wu, H.; Wang, J.; Azzam, H.
(2007).
Modelling retrieval models in a probabilistic relational algebra with a new operator: the relational Bayes.
*The International Journal on Very Large Data Bases (VLDB) 17(1),* pages 5–37.

Suciu, D.; Olteanu, D.; Ré, C.; Koch, C.
(2011).
*Probabilistic Databases.*