The logical view on IR

IR as inference

Example:

- classical IR:
  - $d = \{t_1, t_2, t_3\}$
  - $q = \{t_1, t_3\}$
  - retrieval: $q \subseteq d$?

- 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
Example:
- $d$: 'squares'
- $q$: 'rectangles'
- Thesaurus: 'squares' $\rightarrow$ 'rectangles'

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

d: 'quadrangles'
q: 'rectangles'
⇒ uncertain knowledge required
'quadrangles' 0.3 'rectangles'

[Rijsbergen 86]:
IR as uncertain inference
Retrieval \( \hat{=\text{estimate probability}} \) \( P(d \rightarrow q) = P(q|d) \)

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?
⇒ query cannot be expressed in propositional logic

predicate logic:
\( d: \) tree(t1). house(h1). left(h1,t1).
\( \Rightarrow \) tree(X) & house(Y) & left(X,Y).

Relational Structures: Datalog

Datalog program: finite set of rules each expressing a conjunctive query
\[ t(X_1, \ldots, X_k) : \neg r_1(U_{11}, \ldots, U_{1m_1}), \ldots, r_n(U_{n1}, \ldots, U_{nm_n}) \]
where each variable \( X_i \) occurs in the body of the rule (this way, every rule is safe).

woman(X) :- person(X), sex(X,female).
pth(X,Y) :- link(X,Y).
pth(X,Z):-link(X,Y), pth(Y,Z).

Datalog rules

\[ t(X_1, \ldots, X_k) : \neg r_1(U_{11}, \ldots, U_{1m_1}), \ldots, r_n(U_{n1}, \ldots, U_{nm_n}) \]
corresponds to the logical formula
\[ \forall X_1 \ldots \forall X_k \forall U_{11} \ldots \forall U_{nm_n} \]
\[ t(X_1, \ldots, X_k) \land \neg r_1(U_{11}, \ldots, U_{1m_1}) \land \ldots \land \neg r_n(U_{n1}, \ldots, U_{nm_n}) \]
t\( t(X_1, \ldots, X_k) \) 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 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

Relational Model
Projection

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

Projection: what is the collection about?

\[ \text{topic}(T) := \text{index}(D,T). \]
### Relational Model

#### Selection

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

**Selection:** which documents are about IR?

\[
\text{aboutir}(D) := \text{index}(D,\text{ir}).
\]

#### Join

<table>
<thead>
<tr>
<th>DOCNO</th>
<th>TERM</th>
<th>author</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ir</td>
<td></td>
<td>smith</td>
</tr>
<tr>
<td>1</td>
<td>db</td>
<td></td>
<td>smith</td>
</tr>
<tr>
<td>2</td>
<td>ir</td>
<td></td>
<td>miller</td>
</tr>
<tr>
<td>3</td>
<td>db</td>
<td></td>
<td>johnson</td>
</tr>
<tr>
<td>3</td>
<td>oop</td>
<td></td>
<td>firefly</td>
</tr>
<tr>
<td>4</td>
<td>ir</td>
<td></td>
<td>bradford</td>
</tr>
<tr>
<td>4</td>
<td>ai</td>
<td></td>
<td>firefly</td>
</tr>
<tr>
<td>5</td>
<td>db</td>
<td></td>
<td>bates</td>
</tr>
<tr>
<td>5</td>
<td>oop</td>
<td></td>
<td>bates</td>
</tr>
</tbody>
</table>

**Join:** who writes about IR?

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

#### Union

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

**Union:** which documents are about IR or DB?

\[
\text{irordb}(D) := \text{index}(D,\text{ir}).
\]

\[
\text{irordb}(D) := \text{index}(D,\text{db}).
\]

#### Difference

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

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

\[
\text{irnotdb}(D) := \text{index}(D,\text{ir}) \land \neg \text{index}(D,\text{db}).
\]
The Probabilistic Relational Model

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

index

<table>
<thead>
<tr>
<th>β</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>

aboutdb(D) :- index(D,db).

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

Intensional semantics

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

Method: Event keys and event expressions

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

Extensional vs. intensional semantics

<table>
<thead>
<tr>
<th>link</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>d2</td>
<td>d1</td>
</tr>
</tbody>
</table>

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

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

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

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]

Event keys and event expressions

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

?- about(D,ir) & about(D,db).
gives

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

\[d2 \ [l(d2,d1) & dT(d1,ir) & l(d2,d1) & dT(d1,db)\] 0.7 \cdot 0.9 \cdot 0.5 = 0.315
Recursion

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

\begin{itemize}
\item \text{about}(D,ir)
\item \text{d1} \ [\text{dT}(d1,ir) \mid \text{l}(d1,d2) \& \text{l}(d2,d3) \& \text{l}(d3,d1) \& \text{dT}(d1,ir)]
\item \text{d3} \ [\text{l}(d3,d1) \& \text{dT}(d1,ir)]
\item \text{d2} \ [\text{l}(d2,d3) \& \text{l}(d3,d1) \& \text{dT}(d1,ir)]
\end{itemize}

\[\text{about}(D,db) \& \text{about}(D,ir)
\item \text{d1} \ [\text{dT}(d1,ir) \& \text{dT}(d1,db)]
\item \text{d3} \ [\text{l}(d3,d1) \& \text{dT}(d1,ir) \& \text{l}(d3,d1) \& \text{dT}(d1,db)]
\item \text{d2} \ [\text{l}(d2,d3) \& \text{l}(d3,d1) \& \text{dT}(d1,ir) \& \text{dT}(d1,db)]
\]

Possible worlds semantics

\[\text{0.9 docTerm}(d1,ir).\]
\[P(W_1) = 0.9: \{\text{docTerm}(d1,ir)\}\]
\[P(W_2) = 0.1: \{\}\]

Computation of probabilities for event expressions

1. transformation of expression into disjunctive normal form
2. application of sieve formula:
   \[P(a \lor b) = P(a) + P(b) - P(a \land b)\]
   \[P(c_1 \lor \ldots \lor c_n) = \sum_{1 \leq i_1 < \ldots < i_n \leq n} \sum_{1 \leq i_1 < \ldots < i_n \leq n} P(c_{i_1} \land \ldots \land c_{i_n}).\]
   \[\leadsto\text{exponential complexity}\]
   \[\leadsto\text{use only when necessary for correctness}\]
   \[\text{see [Dalvi & Suciu 07]}\]

Possible interpretations:

\begin{itemize}
\item \text{l1}: \text{P}(W_1) = 0.3: \{\text{docTerm}(d1,ir)\}
\item \text{P}(W_2) = 0.3: \{\text{docTerm}(d1,ir), \text{docTerm}(d1,db)\}
\item \text{P}(W_3) = 0.2: \{\text{docTerm}(d1,db)\}
\item \text{P}(W_4) = 0.2: \{\}\n\item \text{l2}: \text{P}(W_1) = 0.5: \{\text{docTerm}(d1,ir)\}
\item \text{P}(W_2) = 0.1: \{\text{docTerm}(d1,ir), \text{docTerm}(d1,db)\}
\item \text{P}(W_3) = 0.4: \{\text{docTerm}(d1,db)\}
\item \text{l3}: \text{P}(W_1) = 0.1: \{\text{docTerm}(d1,ir)\}
\item \text{P}(W_2) = 0.5: \{\text{docTerm}(d1,ir), \text{docTerm}(d1,db)\}
\item \text{P}(W_3) = 0.4: \{\}\n\end{itemize}

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:

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

Relational Bayes

Example: $P(\text{Nationality} | \text{City})$

```sql
<table>
<thead>
<tr>
<th>nationality_x City</th>
<th>nationality_x City</th>
<th>P(Nationality-City)</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;British&quot;</td>
<td>&quot;London&quot;</td>
<td>0.600</td>
<td>&quot;British&quot;</td>
</tr>
<tr>
<td>&quot;British&quot;</td>
<td>&quot;London&quot;</td>
<td>0.500</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>&quot;British&quot;</td>
<td>&quot;London&quot;</td>
<td>0.250</td>
<td>&quot;Danish&quot;</td>
</tr>
<tr>
<td>&quot;British&quot;</td>
<td>&quot;London&quot;</td>
<td>0.050</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>&quot;British&quot;</td>
<td>&quot;London&quot;</td>
<td>0.000</td>
<td>&quot;French&quot;</td>
</tr>
<tr>
<td>&quot;German&quot;</td>
<td>&quot;Dortmund&quot;</td>
<td>0.033</td>
<td>&quot;Turkish&quot;</td>
</tr>
<tr>
<td>&quot;German&quot;</td>
<td>&quot;Dortmund&quot;</td>
<td>1.000</td>
<td>&quot;Scottish&quot;</td>
</tr>
</tbody>
</table>
```

1. $P(\text{Nationality} | \text{City})$
2. `nationality_x (Nat, City) :-`
3. `nationality_x (Nat, City) | (City);`

Relational Bayes

Example: $P(t|d)$

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

1. $P(t|d)$
2. `P(t|d) SUM(term(DocId) :- term(Term, DocId) | (DocId));`
3. `P(t|d) SUM(term(DocId) :- term(Term, DocId) | (DocId));`
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-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)}\}
?- l-sports(jo) 
P(W_1) + P(W_3) = 0.55

Probabilistic rules
Rules for independent events

sameauthor(D1,D2) :- author(D1,X) & author(D2,X).
0.5 link(D1,D2) :- refer(D1,D2).
0.2 link(D1,D2) :- sameauthor(D1,D2).

?- link(D1,D2) :- refer(D1,D2) & sameauthor(D1,D2).

P(l|r), P(l|s) \rightarrow P(l|r \land s)?

Probabilistic rules
Rules for uncertain facts:

sameauthor(D1,D2) :- author(D1,X) & author(D2,X).
0.5 link(D1,D2) :- refer(D1,D2).
0.2 link(D1,D2) :- sameauthor(D1,D2).

?- link(D1,D2) :- refer(D1,D2) & sameauthor(D1,D2).

P(l|r), P(l|s) \rightarrow P(l|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

refer
\rightarrow
sameauthor
\rightarrow
link
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

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

Vague Predicates in Probabilistic Datalog

\[ [\text{Fuhr & Roelleke 97}] [\text{Fuhr 00}] \]

\[ X \approx Y \]

<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>

Example: Shopping 45 inch LCD TV

\[ \text{vague predicates as builtin predicates:} \]

\[ X \approx Y \]

\[ \text{query(D)}: \] Category(D, tv) \& type(D, lcd) \& size(D, X) \& \( \approx(X, 45) \)
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

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
Aggregation (1)

Who are the major IR authors?

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

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

Aggregation through projection!

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 (2)

Who are the major IR authors?

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

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

irauths SUM(Name) :- irdbauth(Doc,Name) | (Name)

Aggregation through summing:
Expressiveness in XML Retrieval

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
<dc:title>Generic Algebras</dc:title>
...<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>

XML structure: 3. XPath
/doc/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 → 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 $a :=$ avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
<publisher>
  <name> $p/text() </name>
  <avgprice> $a </avgprice>
</publisher>

XML content typing

XML content typing: 1. Text

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

XML content typing: 3. Object Types

Based on Tagging / Named Entity Recognition

- 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
Description Logic/Ontologies

- Disjoint events
- Relational Bayes
- Probabilistic rules
- Thesaurus
- Introduction into OWL
- SPARQL

Thesaurus knowledge:

\[ \text{square} = \text{quadrangle} \land \text{regular-polygon} \]

Description logic

- based on semantic networks
- more expressive than thesauri
  - instances of concepts
  - roles between (instances of) concepts

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
Classes (1)

Class(Female partial Animal)

    <owl:Class rdf:ID="Female">
    <rdfs:subClassOf rdf:resource="#Animal"/>
    </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>

Classes (2)

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 (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>
```

Instances

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

```
<Kain rdf:ID="Kain">
  <hasFather rdf:resource="#Adam"/>
  <hasMother rdf:resource="#Eve"/>
  <shoesize>10</shoesize>
</Male>
```

Property restrictions

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

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

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

▶ *uncertainty*: only deterministic relationships possible, no weighting or probabilistic facts

⇒ "\( \Pr(\text{hasFather(lisa,thomas)})=0.9 \)" 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

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

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

http://www.eecs.qmul.ac.uk/~thor/

Outlook
IR Systems vs. DBMS

Don’t program search engines, design them

http://www.spinque.com/
Towards an IRMS

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.

References II

Fuhr, N.; Lalmas, M.
(2007).
Advances in XML retrieval: the INEX initiative.

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

Fuhr, N.; Röleke, T.
HySpirit – a Probabilistic Inference Engine for Hypermedia Retrieval in Large Databases.

References III

Fuhr, N.
A Probabilistic Framework for Vague Queries and Imprecise Information in Databases.

Fuhr, N.
(2000).
Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications.
Journal of the American Society for Information Science 51(2), pages 95–110.

Lalmas, M.; Roelleke, T.; Fuhr, N.
(2002).
Intelligent Hypermedia Retrieval.

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