Open dialogue management for relational databases

Ben Hixon$^1$ and Rebecca J. Passonneau$^2$

$^1$University of Washington
$^2$Columbia University
Massive Expansion of Information Sources and Interfaces

Relational Databases (RDBs)

Select * from BibRec
where
(author.last='king' &
author.first='stephen' &
pub.date= ...) limit 3;

some methods of access require more work than others
Tradeoffs in Information Access

**RDBs**
- Powerful semantics
- Complex query language
- Requires Expertise

**Ontologies**
- Powerful semantics
- Support extensive inference through inheritance
- Requires Expertise

**Keyword Search**
- Easy for most users
- Users do almost all the WORK:
  - Assemble answers from searches
  - Must draw own inferences

**Dialogue Interfaces**
- Easy for almost all users
- Access with natural dialogue
- Lacks portability: engineering requires domain expertise

---

Select * from BibRec where (author.last='king' & author.first='stephen' & pub.date= ...) limit 3;
What If: RDBs had Reference-Librarian-like Agents?

Relational Databases (RDBs)

What’s Stephen King’s latest book?

Select * from BibRec
where
(author.last='king' &
author.first='stephen' &
pub.date= ...) limit
Portability: dialogue manager bottleneck

User goals and appropriate semantics must be hand-defined

Commercial systems: hand-defined call-flow graphs (Paek & Pieraccini 2008)

Reinforcement-Learned dialogue managers rely on small sets of dialogue states and actions hand-defined by domain experts

Motivation: space of dialogue moves constrained by database
Open Dialogue Management

Generate space of dialogue moves from backend DB

Uses domain-independent heuristics to inspect an RDB and determine potential information needs

Frees users to explore the RDB without prior knowledge of what can be asked

Exploits RDB semantics and contents, e.g. detect natural language fields in rdb and calculate most specific vocabulary
Outline

• Background:
  • Dialogue systems that access databases
  • Dialogue management: deciding what dialogue act to take

• Open dialogue management with ODDMER: 3 modules
  1. Vocabulary selection: Find a common vocabulary
  2. Focus discovery: Infer which tables address basic goals
  3. Focus agents: FSM to manage dialogues for dedicated goal

• Evaluation
  • Three relational databases in different domains
  • Simulated users: control for users’ prior knowledge
Information-seeking dialogues

User goal is to satisfy some information need, e.g.:

– select a tuple from a table: “I want a Stephen King book”
– retrieve the value of an attribute: “Is my flight on time?”, “Who wrote Moby Dick?”
– aggregate over a set of tuples: “How many Italian restaurants in this neighborhood?”
– compare tuple values: “Which restaurant is more expensive?”

Our focus: tuple-selection in slot-filling dialogues
Dialogue manager: user utterance to system action

1. Compute new Dialogue State:
   - User information need: select a book tuple
   - Slots filled: author = Stephen King, date = most recent
   - ...

2. Decide next action. Possibilities:
   - Find answer and inform user
   - Ask user to confirm which Stephen King
   - Fail to find answer, offer alternative information
   - ...

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Database Constrains Dialogue

Our premise: structure and contents of the database constrain types of dialogues users can fruitfully have.

Open dialogue managers compute metaknowledge about their database to pick their states and actions.

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
RW: How RDBs constrain dialogue

  – portable dialogue management
  – automatically cluster attributes to present content summaries

• Demberg and Moore (2006)
  – choose vocabulary according to a user model, e.g. restaurant price for a student
  – need a manually crafted user model, tables of interest predefined

• Hastie, Liu, and Lemon (2009)
  – generate dialogue policies for arbitrary DB tables; rely on a business process model, don’t deal with multi-table databases
ODDMER: Open-Domain Dialogue Manager

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Three Steps

1. Vocabulary selection: attribute-level metaknowledge
   - Vocabulary problem (Furnas et al. 1987): mismatch between user and system vocabulary
   - Solution: ODDMER uses binary classifier for attribute intelligibility, rank by specificity

2. Focus discovery: table-level metaknowledge
   - Which tables give users and system most to discuss? What are the basic user goals for a given rdb?
   - Solution: schema summarization, a random walk algorithm to score tables by verbal information, size, connectivity

3. Focus agents
   - Generate an FSM for each dedicated user goal
Not every attribute is useful: Intelligibility and the Vocabulary Problem

**Vocabulary Problem:** mismatch between system and user vocab

```
S: Do you know the book’s KLAS_ID?
U: No
S: What about the Language Code?
U: I have no idea...
S: Do you at least know the Title?
U: Yes!
```

System shouldn’t use attributes that users won’t know.

**Vocabulary selection:** choosing attributes a user is likely to know

We build a binary classifier to label attributes as *intelligible* or not
Vocabulary selection

task = given a database table, choose the attributes the system should use in a dialogue

want: intelligible attributes the user can readily supply

don’t want: attributes that make user go ‘huh?’

solution: treat as learning problem, build a binary classifier on labeled out-of-domain training data
Attribute intelligibility

4 annotators labeled the Microsoft AdventureWorks Cycling Company database
- 84 attributes total
- 3 out of 4 annotators agreed on 67

These 67 attributes (393,520 total string values, 123,901 unique string values) = training data

Extract features over all the values of each attribute.
Feature extraction and classification

Treat an attribute as a set of string values, extract features over set of values that represent the set

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of unique to total characters in all values</td>
<td></td>
</tr>
<tr>
<td>Mean ratio of unique to total characters per value</td>
<td></td>
</tr>
<tr>
<td>Ratio of numeric to total characters in all values</td>
<td></td>
</tr>
<tr>
<td>Ratio of unique to total values</td>
<td></td>
</tr>
<tr>
<td>Ratio of unique to total words in all values</td>
<td></td>
</tr>
<tr>
<td>Total number of characters in all values</td>
<td></td>
</tr>
</tbody>
</table>

Features engineered to be domain independent and to capture the linguistic expressiveness of the values.

RIPPER rule-learning algorithm gives classifier with 77% precision, 78% recall with 10-fold cross-validation.
Attributes differ in specificity

U: I’d like a Stephen King book
U: No
U: No
U: Pet Sematary?

S: He has more than 100 books. How about Carrie?
S: Do you want The Stand?
S: Just give me a title!
S: Okay, that’s available.

Author attribute: less specific
Title attribute: more specific

We can calculate attribute specificity given the database

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Semantic specificity

how uniquely does the attribute describe the entity?

| ... | I want a book by Stephen King |
| U   | Would you like Carrie?       |
| U   | No                           |
| S   | Would you like The Stand?    |
| U   | No                           |
| S   | Would you like The Shining?  |
| U   | Yes                          |

| ... | I want a book by Harper Lee |
| U   | Would you like To Kill a Mockingbird? |
| S   | Yes                          |

semantic specificity: between 0 and 1, scores how unambiguously its values map to rows in the table
Intelligible attributes for BOOK

<table>
<thead>
<tr>
<th>Intelligible Attributes</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANNOTATION</td>
<td>0.958</td>
</tr>
<tr>
<td>TITLE</td>
<td>0.878</td>
</tr>
<tr>
<td>SORTTITLE</td>
<td>0.878</td>
</tr>
<tr>
<td>AUTHOR</td>
<td>0.300</td>
</tr>
<tr>
<td>NARRATOR</td>
<td>0.018</td>
</tr>
<tr>
<td>PUBLISHER</td>
<td>0.016</td>
</tr>
<tr>
<td>SERIES</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Out of 32 attributes, only 8 in the book table are intelligible.
Not every table is interesting

Most users want books from this table

Nobody cares about this table

The Heiskell Library database schema

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Focus discovery

In tuple-selection dialogues on RDBs, each table corresponds to a distinct **dialogue focus**

**Task:** Choose tables of most interest to user

**Solution:** use schema summarization to rank tables by importance as a function of **size**, **connectivity**, and **verbal information**

---

**Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation**

---

S: Do you want a Circulation Transaction today?

U: Uh, no…I just want a book.
Schema summarization

**Input**: large multi-table database

**Output**: tables ranked by summary score.

Random walk algorithm

- inspired by work by Yang et al (VLDB 2009): score tables by size, attribute entropy, connectivity

- our changes for dialogue: score the flow of verbal information over the joins
Database schema

Database schema = an undirected graph \( G = <R,E> \)
- nodes \( r \) in \( R \): tables in the DB
- edges \( e \) in \( E \): joins between tables

Schema summary of size \( k \): the \( k \) most important nodes in the schema

Table Importance?
- for Yang: size, entropy of its attributes, connectivity with other tables
- but: not all attributes used in dialogue. unintelligible attribs shouldn’t contribute to table importance
- we look at the verbal information content
Verbal information content

We build a transition matrix for every pair of tables in the DB.

Initialize each table’s verbality score to its verbal information content \( V(T) \):

\[
V(T) = \log(|T|) + \sum_{a \in A'} H(a)
\]

\( A' \) = the set of intelligible attributes in the table

\( H(a) \) is the entropy of each attribute:

\[
H(a) = -\sum_{k \in K} p_k \log(p_k)
\]

Intuition: without considering connectivity, more important tables have more tuples and diverse intelligible attributes.
Incorporate connectivity

Find flow of verbal information over table joins. Information transfer over joins given by:

\[ IT(j) = \frac{H(j)}{V(T) + \sum_{a \in A} q_a H(a)} \]

Intuition: information transfer is information of the join over the verbal information of the table and information over all joins.

\[ j = \text{join attribute} \]
\[ q_a = \text{number of joins a belongs to} \]
Transition matrix for a dialogue database schema

Let $P(T,R) = \text{transition probability between two tables } T \text{ and } R$.

If $T \neq R$, then $P(T,R)$ is total information transfer between $T$ and $R$:

$$P(T, R) = \sum_{j \in J} IT(j)$$

where $J$ is all joins between tables $T$ and $R$.

Diagonal entries: how likely information stays in each table:

$$P(T, T) = 1 - \sum_{T \neq R} P(T, R)$$
Verbal information and Verbality

Verbality of table $T_i$: the $i^{th}$ element of the stable distribution of a random walk over the transition matrix

<table>
<thead>
<tr>
<th>T</th>
<th>V(T)</th>
<th>Verbality(T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>88.2</td>
<td>45.4</td>
</tr>
<tr>
<td>Heading</td>
<td>31.7</td>
<td>22.4</td>
</tr>
<tr>
<td>BibHeadingLink</td>
<td>19.2</td>
<td>23.6</td>
</tr>
<tr>
<td>CirculationHistory</td>
<td>17.9</td>
<td>24.9</td>
</tr>
<tr>
<td>Holding Stats</td>
<td>15.2</td>
<td>24.0</td>
</tr>
<tr>
<td>Patron Properties</td>
<td>12.7</td>
<td>21.9</td>
</tr>
<tr>
<td>Reserve</td>
<td>12.2</td>
<td>25.5</td>
</tr>
<tr>
<td>Patron</td>
<td>9.0</td>
<td>18.6</td>
</tr>
</tbody>
</table>
Focus Agent Generation

Agent-based approach to dialogue management (Bohus&Rudnicky 2009, Nguyen&Wobcke 2005)

Root agent: (1) begins dialogue, (2) presents schema summary, (3) determines user need, (3) launches goal-specific agent

Focus agents responsible for dedicated user goal:
- FSM constructed from intelligible attributes
- current implementation system-initiative
- prompts for intelligible attributes ordered by specificity
Evaluation

2 simulated users, C and L


Realistically limited knowledge. Like a human user, L has a vocabulary problem.

S: Do you know the PatronObj? C: Of course I do, it’s 102294.5232
S: Do you know the BibHeadingLinkObj? L: No idea what you’re talking about

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Simulating the vocabulary problem

Simulate L’s realistic knowledge by assigning different probabilities to different attributes. L more likely to know book’s title than its ISBN.

Need method robust to many missing values. Use relative occurrence in Gigaword as L’s likelihood of knowing an attribute.

Produced reasonable results, e.g.:
  Title: 100% likelihood
  Author: 78% likelihood
  Place Published: 75% likelihood
  ISBN: 0% likelihood

Similar to Selfridge and Heeman (2010) who also simulate users with different knowledge levels. Their users don’t know different attributes with different likelihoods.
### Dialogue lengths on 3 databases

<table>
<thead>
<tr>
<th></th>
<th>Heiskell</th>
<th>Grocery</th>
<th>Eve</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/N/R</td>
<td>15.9 ± 0.4</td>
<td>11.1 ± 0.2</td>
<td>16.3 ± 0.4</td>
</tr>
<tr>
<td>C/N/S</td>
<td>9.0 ± 0.0</td>
<td>9.0 ± 0.0</td>
<td>9.0 ± 0.0</td>
</tr>
<tr>
<td>C/V/R</td>
<td>11.5 ± 0.2</td>
<td>11.1 ± 0.3</td>
<td>10.6 ± 0.1</td>
</tr>
<tr>
<td>C/V/S</td>
<td>9.5 ± 0.1</td>
<td>9.0 ± 0.0</td>
<td>9.0 ± 0.0</td>
</tr>
<tr>
<td>L/N/R</td>
<td>25.3 ± 0.8</td>
<td>13.1 ± 0.2</td>
<td>17.7 ± 0.5</td>
</tr>
<tr>
<td>L/N/S</td>
<td>16.5 ± 0.6</td>
<td>9.3 ± 0.1</td>
<td>9.4 ± 0.1</td>
</tr>
<tr>
<td>L/V/R</td>
<td>12.4 ± 0.2</td>
<td>12.2 ± 0.1</td>
<td>10.7 ± 0.1</td>
</tr>
<tr>
<td>L/V/S</td>
<td>11.3 ± 0.2</td>
<td>9.7 ± 0.1</td>
<td>9.0 ± 0.0</td>
</tr>
</tbody>
</table>

- **C** = complete-knowledge
- **N** = no vocabulary selection
- **S** = ordered by specificity
- **L** = incomplete-knowledge
- **V** = Intelligible vocabulary
- **R** = ordered randomly
Vocabulary selection gives limited-knowledge users efficient dialogues

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Dialogue Length (Heiskell)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/N/R</td>
<td>15.9 ± 0.4</td>
</tr>
<tr>
<td>C/N/S</td>
<td>9.0 ± 0.0</td>
</tr>
<tr>
<td>C/V/R</td>
<td>11.5 ± 0.2</td>
</tr>
<tr>
<td>C/V/S</td>
<td>9.5 ± 0.1</td>
</tr>
<tr>
<td>L/N/R</td>
<td>25.3 ± 0.8</td>
</tr>
<tr>
<td>L/N/S</td>
<td>16.5 ± 0.6</td>
</tr>
<tr>
<td>L/V/R</td>
<td>12.4 ± 0.2</td>
</tr>
<tr>
<td>L/V/S</td>
<td>11.3 ± 0.2</td>
</tr>
</tbody>
</table>

Average dialogue length over 1000 simulated dialogues.

Dialogues continue until a tuple is successfully ordered.

C doesn't get any benefit from vocabulary selection: dialogues already at minimum length just from ordering by specificity.

Vocabulary selection and order-by-specificity big help to L. With both together, L’s dialogues almost as fast as C’s dialogues.

Dialogue length decreases for limited knowledge users with vocabulary selection and order-by-specificity.

Background • ODDMER • Vocabulary selection • Focus discovery • Evaluation
Conclusion

We find each table’s useful attributes by calculating their intelligibility and specificity.

We use schema summarization to choose the most important tables and present them to the user.

Evaluation with simulated users shows that specific intelligible vocabulary produces shorter dialogues.

Without domain expertise or a human in the loop, the database itself can constrain dialogue management.
Future work

Much more metaknowledge to be extracted from database structure and contents

move beyond tuple selection to extract arbitrary information needs: inspired by DB literature for automatically computing most likely SQL queries for an arbitrary database (Jayapandian & Jagadish, 2008)

ODDMER bound by table and attribute names: can we infer more meaningful labels?

we don’t consider optimizing dialogue strategies: can open dialogue management preprocess state space for reinforcement learning?

Evaluate on more databases and with real users
Take-home message

What’s in the database and how it’s organized constrain the space of fruitful dialogues.

These constraints can be extracted as metaknowledge in a domain-independent manner.
Thanks!

Special thank you to: Susan Epstein, Oren Etzioni, Luis Gravano, Wlodek Zadrozny, our annotators at Columbia University, and our anonymous reviewers