Explaining Structured Queries in Natural Language

-- Yunjia Zhou
ROADMAP

• Introduction
• Query Representation
• Capturing Query Semantics
• Query Translation
• Experiments
ROADMAP

- Introduction
  - Motivation
  - Main Contribution
- Query Representation
- Capturing Query Semantics
- Query Translation
- Experiments
Motivation

• Provide user with a textual explanation of the Structured Queries
  – Assist user in forming queries correctly
  – Help user to identify failure parts of the query when return error answers
Goal: Translate from SQL query to NL

• Given a query $q$ over a database $D$, generate a narrative that captures the intended meaning or objective of $q$.

• Challenges:
  • Insufficient SQL semantics
  • Complexity of the queries
  • Equivalent alternative expressions of a query
Idea

• Graph-based approach
  – Represent structured queries as directed graphs

• Extensible template mechanism
  – Annotate the graph elements with labels

• Three translation strategies
  – BST algorithm
  – MRP algorithm
  – TMT algorithm
Contributions

• Introduced a novel query graph model
• Gave semantics to a query by annotating the query graph edges with template labels using Extensible template mechanism
• Presented different graph traversal strategies
• Presented an algorithm for templates selection
• Compared translation algorithms
Database Graph

• A database $D$ is represented by its database graph $G(V,E)$

• Nodes in V are
  — Relation nodes, R
  — Attributes nodes, A

• Edges in E are
  — Membership edges, $E^\mu$
  — Selection edges, $E^\sigma$
  — Predicate edges, $E^\theta$
Database Graph

• An example

Departments(DepID, DepCode, Name)  Courses(CourseID, DepID, Title)
Instructors(InstrID, Name)  Students(SuID, Name, Class, GPA)
CourseSched(CourseID, Year, Term, InstrID, TimeSlot)
StudentHistory(SuID, CourseID, Year, Term, Grade)
Comments(SuID, CourseID, Year, Term, Text, Rating, Date)

Fig. 1. An example course database

Fig. 2. A join on database graph
Query Graphs

- SPJ queries, an extension of the database graph
- Nodes
  - Relation nodes
  - Attribute nodes
  - Value nodes: one for each value or a set of values specified in the query qualification
Query Graphs

• Edges
  – Membership edges
  ➢ Predicate edges
    ➢ A θ Ω
    ➢ Ω: a single value or a set of values or an attribute
    ➢ Θ: comparison operator (e.g., =, <, >, <> and Like)
    ➢ If Ω is value, θ is a selection predicate edge; If Ω is an attribute, θ is a join predicate edge
  – Selection edges
Query Graphs

• An SPJ example

```sql
select s.name, s.GPA, c.title, i.name, co.text
from students s, comments co
    studenthistory h, courses c, departments d,
    coursesched cs, instructors i,
where s.suid = co.suid and
    s.suid = h.suid and h.courseid = c.courseid and
    c.depid = d.depid and
    c.courseid = cs.courseid and cs.instrid = i.instrid and
    s.class = 2011 and co.rating > 3 and
    cs.term = 'spring' and d.name = 'CS'
```

Fig. 3. A SPJ query
Query Graphs

• Extension
  – To capture functions, expressions and renaming operations as well as order-by, group-by, and having clauses

Example 2 Let us consider the following query.

```sql
select year, term, max(grade)
from studenthistory
group by year, term having avg(grade) > 3
```

(a) A group-by query
Query Graphs

• Extention
  – Nested queries (subquery)

Example 3 Let us consider the following query.

```sql
select s.name from students s
where NOT EXISTS (select * from students s2
  where s2.GPA > s.GPA)
```

(b) A nested query
ROADMAP

• Introduction
• Query Representation
  ▮ Capturing Query Semantics
  ▮ Labels
  ▮ Templates
• Query Translation
• Experiments
Capture Query Semantics

• A template mechanism that allows us to represent semantics of query graph elements
• Labels
• Templates
Capture Query Semantics

• Labels:
  – stored on the database graph for both nodes and edges
  – Node
    • Each node has a conceptual meaning.
    • E.g., relation $Students$ – “students”, function $max$ – “the greatest”
  – Edge
    • Each edge is also annotated by a label
Capture Query Semantics

Label
Capturing Query Semantics

- Templates
  - Define template labels at different granularity levels
  - Provide an extensible template mechanism to fuse these template
  - generic template: database agnostic

\[ l((v, u)) = expr_1 + l(v) + expr_2 + l(u) + expr_3 \]
ROADMAP

• Introduction
• Query Representation
• Capturing Query Semantics
  ➢ Query Translation
    ➢ Query Subject
    ➢ Query graph traversal
    ➢ Template Selection
    ➢ Discussion
• Experiments
Query Translation

• Selection of query subject
  – Usually a relation with attributes projected in the select-clause
  – Definition 1: Primary relation Rp
    • A relation storing information for a set of entities of the same name type, e.g., Students
    • Candidates for query subject
  – Definition 2: Secondary relation Rs
    • A relation stores information for a relationship of entities that are stored in different relations, e.g., StudentHistory
Query Translation

• Select query subject
  – Primary relation
  – Select “central” in the query graph
  – QSuB Algorithm
    • Find shortest longest path

\[
\max_{R_x \in \mathbb{R}} (\delta(R_q, R_x)) \leq \{ \max_{R_x \in \mathbb{R}} (\delta(R_i, R_x)) : \forall R_i \in \mathbb{R} \}
\]
Query Translation
Query Translation

• Query graph traversal
  – BST Algorithm
• DFS-like query translation

“Find the title of courses, the name of instructors, the gpa and name of students, and the description of comments for courses that are taught by instructors, are taken by students that gave comments, and are offered by departments. Return results only for courses whose term is spring, students whose class is 2011, comments whose rating is greater than 3, and departments whose name is CS.”
Query Translation

• MRP Algorithm:
  – Reference Point
    Definition: too long

Before
“Find the names of students and the titles of the courses taken by these students and the names of the instructors that taught courses taken by these students”

After
“Find the names of students and the titles of the courses taken by these students and the names of the instructors that taught these courses”
Query Translation

• Template Selection
  – To get the best combination of generic and specific templates
    • Composeable templates
    • Apply as few templates as possible
  – TS Algorithm
  – TMT Algorithm
ROADMAP

• Introduction
• Query Representation
• Capturing Query Semantics
• Query Translation

➤ Experiments
  ➤ Effectiveness
  ➤ Performance
Experiments

• From SQL to NL
• From NL to SQL
Thank you!