Keyword Search

Using Keyword Searches in Databases

Problems

- How can we find data without knowing queries or schemas?
- How do we find the relevant information to a search?
- Complexity of query languages makes it hard for the average user.
- Additionally, data is usually spread across several tables.

One Possible Solution

- Database is modeled as a graph
  - Nodes are the tuples
  - Edges are the references between tuples

The Graph Solution Continued

- Rooted, directed tree connecting keyword nodes
- May include internal nodes that contain no keywords
- Root node has special significance
- May be restricted to relations representing entities
- Avoid relations representing relationships, e.g., "writes"
- Multiple answers may exist
- Ranked by proximity and prestige

The BANKS Algorithm

- Proximity
  - Forward edges are relation that are foreign key → primary key
  - Weight of forward edge is based on schema (how tables are correlated)
  - Backward edges are added to account for "hubs"
  - Weight of backward edge \( u \rightarrow v \propto \text{indegree of } u \)
- Prestige
  - Calculated from indegree of the node
  - Answer tree relevance
  - Edge score \( E = 1 / \Sigma \text{edge-weights} \)
  - Node score \( N = \Sigma \text{root- and leaf-node-weights} \)
  - Ignore weights of internal nodes
  - Normalize and combine using weighting factor \( \lambda \)
  - Additive: \( (1- \lambda) E + \lambda N \)
  - Multiplicative: \( E \lambda N \)

Proximity Example

- Weight of forward edge based on schema
- Weight of backward edge = indegree of edges pointing to the node
Searching for an Answer Tree

- Backward Expanding Search Algorithm:
  - Start at nodes that contain keyword queries.
  - Run concurrent single source shortest path algorithm from these nodes.
  - Create an iterator for each node matching a keyword
    - Traverse the graph edges in reverse direction
  - Output a node whenever it is on the intersection of the sets of nodes reached from each keyword
  - Answers may not be in the most relevant order

An Example

- Query: “sudarshan roy”
  - paper
  - MultiQuery Optimization
  - writes
  - authors S. Sudarshan Prasan Roy

Visualization of BANKS

Using the Banks System

- BANKS provides keyword search coupled with extensive browsing facilities
- Implemented using Java
- Keyword search response times takes about 1 to 3 seconds on
  - DBLP database with 100K tuples and 300K edges
  - P3 600 MHz, 512 MB RAM
- Try it out at www.cse.iitb.ac.in/banks/