Keyword Query Semantics

Keywords are:

- in same tuple
- in same relation
- in tuples connected through primary-foreign key relationships

Score of result:

- distance of keywords within a tuple
- distance between keywords in terms of primary-foreign key connections
- IR-style score of result tree

[1]

Result of Keyword Query

Result is tree *T* of tuples where:

- each edge corresponds to a primary-foreign key relationship
- no tuple of *T* is redundant (minimality)
- - "AND" query semantics: Every query keyword appears in *T*

  - "OR" query semantics: Some query keywords might be missing from *T*

[1]

Shortcomings of Prior Work

- Simplistic ranking methods (e.g., based only on size of connecting tree), ignoring well-studied IR ranking strategies
- No straightforward extension to improve efficiency by returning just top-k results
- Not good in handling free-text attributes

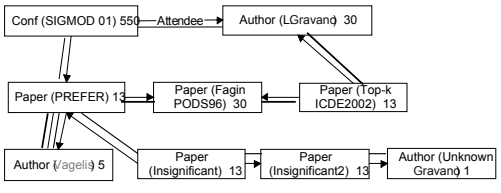
[DBXplorer,DISCOVER]

[1]

Efficient IR-Style Keyword Search over Relational Databases

- Ranking functions
  - Modern RDBMSs already include IR-Style relevance ranking functionality over individual text attributes, which can be exploited to define a ranking scheme.
  - DBXplorer, DISCOVER, and BANKS require all words in the query to be present or else the query receives a score of zero.
  - Other methods calculate ALL results.
  - New method: combine the scores of the individual keywords in a horribly complicated algorithm.

Example

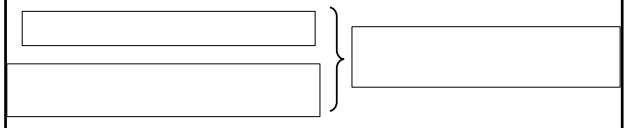


Results

- R1: Vagelis – PREFER – SIGMOD 01 – L.Gravano
- R2: Vagelis – PREFER – Fagin PODS96 – Top-k ICDE2002 – L.Gravano
- R3: Vagelis – PREFER – Insignificant1 paper – Insignificant2 paper – Gravano

[2]

Calculating the 'watermark' for use in picking the top-k results. The ranking algorithm is of similar complexity.

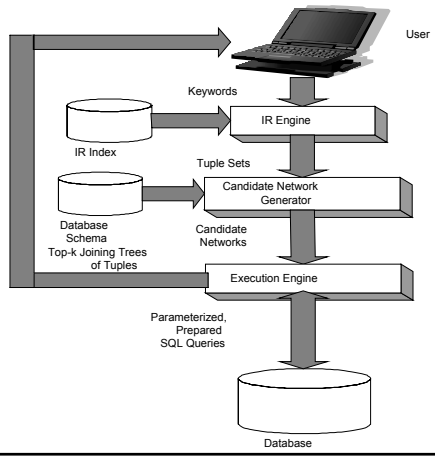


$$A_i(t') = \begin{cases} min(\frac{f_v(t') - \sum_{j<=i}^{k} v_j m_j}{v_i}, M_i) & q_i > v_i <> 0 \\ M_i & q_i > v_i = 0 \\ 0 & q_i = v_i \\ max(\frac{f_v(t') - \sum_{j<=i}^{k} v_j M_j}{v_i}, m_i) & q_i < v_i \end{cases}$$

[3]

Efficient IR-Style Keyword Search over Relational Databases

- Architecture
  - Attempt to rely on existing RDBMS components such as Oracle Text.
  - The Candidate Network tends to be exponential with query size.
  - The Candidate Network Generator is almost identical to that of the DISCOVER method.



[1]

Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Naïve
  - Sparse
  - Single Pipelined
  - Global Pipelined
  - Hybrid

Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Naïve
    - The Execution Engine issues one SQL query for each Candidate Network.
    - The results are sorted to pick the top-k.
    - If the query is a of Boolean-AND semantics, the algorithm will filter out results that do not contain all keywords
    - This approach is similar to the approach taken in DBXplorer and DISCOVER.

Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Sparse
    - Rather than exhaustively processing every CN the Sparse algorithm attempts to improve processing time by eliminating unprocessed CN's that are guaranteed to not produce a top-k result.
      - Computes the maximum hypothetical score
    - Most efficient on queries that produce few results. (low ratio of results to candidate networks)

Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Single Pipelined
    - Calculates Maximum Possible Future Score (MPFS) during preprocessing for use in filtering and ordering.
      - MPFS calculation is actually just an attempt at a tight upper bound
      - As results come in the MPFS is updated
    - If the query is a of Boolean-AND semantics, the algorithm will filter out results that do not contain all keywords
    - Not efficient when used separately for each Candidate Network. However, the algorithm is the main building block of the Global Pipelined algorithm.

Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Global Pipelined
    - Each Candidate Network is evaluated concurrently using a priority preemptive round robin protocol.
      - Each CN is evaluated in its own process with the priority being assigned by its MPFS value.
      - The MPFS is updated fairly regularly.
        - More promising sets finish
        - Less promising sets are removed
    - Most efficient at queries that produce many results

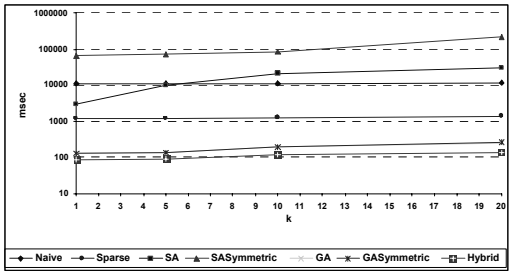
Efficient IR-Style Keyword Search over Relational Databases

- Execution Algorithms
  - Hybrid
    - Estimate the expected number of results
      - If few results are expected, use the Sparse algorithm
      - If many results are expected, use the Global Pipelined algorithm
    - For 'OR' queries, the adding the RDBMS's result size estimates is fairly accurate
    - For 'AND' queries independent results should be eliminated from the 'OR' result set.

Efficient IR-Style Keyword Search over Relational Databases

- Experimental Results
  - Naïve
  - Sparse
  - Single Pipelined (SA)
  - Global Pipelined (GA)
  - SASymmetric
  - GASymmetric
  - Hybrid

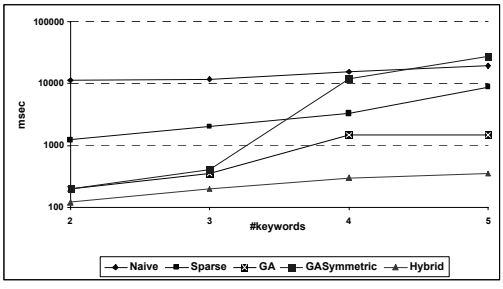
OR Semantics: Effect of Number of Objects Requested *k*



Average execution time of 100 2-keyword queries with maximum candidate-network size of 6

[1]

OR Semantics: Effect of Number of Query Keywords



Average execution time of 100 top-10 queries with maximum candidate-network size of 6

[1]

References

[1] V. Hristidis, L. Gravano, Y. Papakonstantinou

"Efficient IR-Style Keyword Search over Relational Databases" In VLDB 2003.

[2] V. Hristidis Random Walk in Stuctured Databases UCSD Research Seminar, 2003

[3] V. Hristidis, N. Koudas, Y. Papakonstantinou

"PREFER: A System for the Efficient Execution of Multi-Parametric Ranked Queries ".

(ps) In ACM SIGMOD 2001.