Query Relaxation
Using Malleable Schemas
Table Of Contents

1. Introduction
2. Limitations
3. Query Relaxation
4. Implementation Issues
5. Experiments
6. Conclusion
1. Introduction - Motivation

- Clean Schema
- Malleable Schema
- Unstructured data sources
- Structured data sources
1. Introduction - Motivation

xml is the standard for data exchange

xml search

body
title

author

first_name

sur_name

Person

John Gary

name

Pan

x

False

ISA-book

my paper

subject

email

attachment

contents

date

25.03.2006

Dear Sergey,
Please find the attached paper

... ...

Desktop Search

We have many
... ...

Doc

ISA-paper

True
1. Introduction - Goals / Aims

- Query Relaxation is for querying vaguely structured data by using malleable schemas.
- We can relax queries in order to find more results.
  - But also useful to find the correct result, when the original query would not retrieve it.
- The proposed approach furthermore ranks results.
  - According to the probability of satisfying the user's search.
- Combining database queries and Information Retrieval technologies.
1. Introduction - Problems / Challenges

• Challenge 1
  • Example
    - Q1: Select Person Where first-name = "Daniel"
      needs to be relaxed to
    - Q2: Select Person Where first-name = "Daniel"
    - OR name Contains "Daniel"

• We need to know that first-name is part of name

• Challenge 2: Find correlated schema elements, quantify these correlations
2. Limitations

- Entity-Relationship data model
  - Relations involve only two entities and have no attributes
- Malleable schema isn't concise
- For simplicity: containment only, conjunctions only
- Example
  - Q3 : Select Doc As E1 Where E1.title Contains “XML Query” And E1.ISA-paper Contains “True” And E1.author Contains E2 And E2.name Contains “Daniel”
3. Query Relaxation

- Relaxation

- $Q_3 = \{ E_1 \mid E_1.title \ni 'XML' \land E_1.title \ni 'Query' \land E_1.ISA - paper \ni 'True' \land E_1.author \ni E_2 \land E_2.name \ni 'Daniel' \}$

- can be relaxed to

- $Q_5 = \{ E_1 \mid E_1.title \ni 'XML' \land E_1.title \ni 'Query' \land E_1.ISA - paper \ni 'True' \land E_1.author \ni E_2 \land E_2.first - name \ni 'Daniel' \}$

- $Q_6 = \{ E_1 \mid E_1.subject \ni 'XML' \land E_1.subject \ni 'Query' \land E_1.ISA - paper \ni 'True' \land E_1.writer \ni E_2 \land E_2.name \ni 'Daniel' \}$
3. Query Relaxation

• Sorting results by their probabilities of relevance

  • Conditional probability
  \[ P(\text{cancer}|\text{smoker}) = \frac{P(\text{smoker} \cap \text{cancer})}{P(\text{smoker})} \]

  • \( P(Q3|Q5) > P(Q3|Q6) \)

  • Computation is slow, can make independence assumptions to make it feasible
3. Query Relaxation

- Discovering Correlations and Duplicates

  - Brute force approach for estimating \( P(A|A') \)
    - Terms in attribute "first-name" also appear in attribute "name"
    - \( P(\text{name}|\text{first-name}) = 1 \)
    - \( P(\text{first-name}|\text{name}) = 0.5 \)

- Problem
  - Person entity with attributes "first-name" and "surname" in real-world will not have any entry in an attribute called "name"

- Conclusion
  - We need to find duplicates
3. Query Relaxation

- Discovering Correlations and Duplicates

<table>
<thead>
<tr>
<th></th>
<th>title</th>
<th>subject</th>
<th>author</th>
<th>writer</th>
<th>pub-date</th>
<th>rec-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>XML</td>
<td></td>
<td>Daniel</td>
<td></td>
<td>Jan 1999</td>
<td></td>
</tr>
<tr>
<td>E2</td>
<td>XML</td>
<td></td>
<td>Daniel</td>
<td></td>
<td>Dec 2003</td>
<td></td>
</tr>
<tr>
<td>E3</td>
<td>DB</td>
<td></td>
<td>Ullman</td>
<td></td>
<td>Jul 1994</td>
<td></td>
</tr>
<tr>
<td>E4</td>
<td>DB</td>
<td></td>
<td>Ullman</td>
<td></td>
<td>Nov 2001</td>
<td></td>
</tr>
<tr>
<td>E5</td>
<td>AI</td>
<td></td>
<td>Stuart</td>
<td></td>
<td>Nov 2001</td>
<td></td>
</tr>
<tr>
<td>E6</td>
<td>Logic</td>
<td></td>
<td>Stuart</td>
<td></td>
<td>Nov 2001</td>
<td></td>
</tr>
</tbody>
</table>
3. Query Relaxation

• The DSCD Algorithm
  • Ignore relationships between entities
  • Consider only attribute correlations
  • Procedure
    - 1.) Duplicate detection
      • Based on current schema correlations, find all possible duplicates
    - 2.) Correlation detection
      • Based on current duplicates, reassess the schema correlations
    - 3.) If the schema correlations did not change in step 2, stop the process. Otherwise go to step 1
4. Implementation Issues

- Given $n$ entities, $m$ attributes
  - $n^2$ possible pairs of duplicates, $m^2$ possible pairs of attributes
- This is infeasible, so we use preselection
- Procedure of identifying and quantifying correlations:
  - 1. Preparation: Find all possible correlated pairs of attributes
  - 2. Verification: Perform DSCD algorithm
  - 3. Quantification: Identify more duplicates, use them for quantifying schema correlations
5. Experiments – Dataset and Setup

• Combining real world data from different sources
  • Movies provided by www.imdb.com
  • DVD/video items crawled from www.amazon.com
  • Both Describing similar data in different ways
  • Attributes very different but closely correlated

• Used System
  • Implementation in Java 5
  • PC with 2.7GHz CPU and 1GB RAM
  • MySQL 4.1.11
5. Experiments – Running Algorithm

• Preselection
  • Assuming no correlation between attributes of different types
  • Randomly picked up 1000 IMDB entries
  • Compared with Amazon entities using TF-IDF
  • Selected 100 most similar pairs as duplicate candidates
  • Most important and representative attributes used only

• Verification
  • Computed a confidence value for each schema correlation indicating the extent of belief that the match is correct
5. Experiments – Running Algorithm

• Quantification
  • Used K-Mean algorithm to automatically cluster the duplicate candidates into true and false duplicates, based on their confidence values
  • Set the median of the two clusters as a threshold for duplicate detection
  • Found 200 more duplicates that have confidence larger than the threshold
  • Calculated requested probabilities
5. Experiments – Results

|       | Amazon | IMDB    | confidence | P(A|I)  |
|-------|--------|---------|------------|--------|
| 1     | Title  | title   | 0.701      | 0.619  |
| 2     | Actors | actors  | 0.655      | 0.587  |
| 3     | Directors | directors | 0.642  | 0.753  |
| 4     | Languages | languages | 0.382  | 0.711  |
| 5     | Edit~Review | keywords | 0.132  | 0.086  |
| 6     | Directors | producers | 0.102  | 0.072  |
| 7     | Title  | akatitles | 0.090  | 0.097  |
| 8     | ReleaseDate | year     | 0.081  | 0.098  |
| 9     | Directors | writers  | 0.080  | 0.173  |
| 10    | Actors | misc     | 0.072  | 0.047  |
| 11    | Edit~Review | plots    | 0.067  | 0.076  |
| 12    | Actors | writers  | 0.061  | 0.098  |
| 13    | Directors | proddesigners | 0.059  | 0.002  |
| 14    | Directors | cinematographers | 0.054  | 0.023  |
| 15    | Title  | movielines | 0.049  | 0.050  |
| 16    | Edit~Review | taglines | 0.046  | 0.056  |
| 17    | Actors  | producers | 0.046  | 0.072  |
| 18    | Actors  | directors | 0.043  | 0.094  |
| 19    | Audi~Rating | certificates | 0.042  | 0.136  |
| 20    | Actors  | composers | 0.042  | 0.056  |
5. Experiments – Results

| IMDB     | Amazon    | confidence | P(I|A) |
|----------|-----------|------------|------|
| 1        | title     | Title      | 1.000| 0.923|
| 2        | actors    | Actors     | 0.794| 0.720|
| 3        | directors | Directors  | 0.666| 0.792|
| 4        | akatitles | Title      | 0.380| 0.303|
| 5        | languages | Languages  | 0.348| 0.621|
| 6        | distributors | Publisher | 0.264| 0.335|
| 7        | distributors | Manufacturer | 0.264| 0.335|
| 8        | distributors | Label     | 0.264| 0.335|
| 9        | distributors | Studio    | 0.264| 0.335|
| 10       | movielinks | Title      | 0.262| 0.296|
| 11       | producers | Directors  | 0.152| 0.186|
| 12       | writers   | Directors  | 0.104| 0.259|
| 13       | year      | ReleaseDate| 0.081| 0.098|
| 14       | technical | AspectRatio| 0.073| 0.108|
| 15       | actors    | Creators   | 0.067| 0.063|
| 16       | actors    | Directors  | 0.066| 0.135|
| 17       | proddesigners | Directors | 0.059| 0.003|
| 18       | certificates | Audi～Rating| 0.058| 0.220|
| 19       | cinematographers | Directors | 0.058| 0.029|
| 20       | plots     | Edit～Review| 0.036| 0.052|
5. Experiments – Result Quality

- Sensitivity to true duplicates
6. Conclusion

- Query relaxation effective search using malleable schemas
- Probabilistic model
- DSCD Algorithm
- Experimental study using real dataset
- For future research
  - storage management, query interface and flexible mechanism for updating malleable schemas