Similarity of XML Schema Fragments Based on XML Data Statistics

Irena Mlynkova, Jaroslav Pokorny
{irena.mlynkova,jaroslav.pokorny}@mff.cuni.cz

Charles University
Faculty of Mathematics and Physics
Department of Software Engineering
Prague, Czech Republic
Introduction

- XML = a standard for data representation and manipulation
- Growing demand for efficient managing and processing of XML data
  ⇒ Possible optimization: To exploit similarity of XML data
  - We can manage similar data in a similar way
  - We can extend verified approaches to all similar cases
- The amount of approaches is significant
  - A space for further improvements
Goals of This Presentation

Proposal of a similarity function designed for enhancing of XML-to-relational mappings

• Overview of existing approaches
• Proposal of improvement – focus on:
  • Structural similarity
  • Realistic tuning of weights and parameters
• Experiments
• Conclusion
Content

1. Overview of existing approaches
2. Proposed improvement
3. Experiments
4. Conclusion
Approaches to XML Similarity (1)

- **Similarity of documents** $D_1$ and $D_2$
  - How difficult is to transform $D_1$ into $D_2$
    - Tree edit distance
  - A simple representation of $D_1$ and $D_2$ enabling easier comparison
    - e.g. set of paths, document signal

- **Similarity of document** $D$ and **schema** $S$
  - Number of documents which appear in $D$ but not in $S$ and vice versa
    - Common, plus, minus elements
  - The closest tree edit distance between $D$ and all documents valid against $S$
    - Construction of automaton / grammar of $S$
Approaches to XML Similarity (2)

- Similarity of schemes $S_1$ and $S_2$
- Exploitation and combination of supplemental information
  - Predefined similarity rules, similarity of element / attribute names, equality of data types, schema instances, thesauri, previous results, …
- Emphasis on semantic similarity
  - Exploitation: schema-integration systems, dissemination based systems, …
  - Problem: For XML-to-relational mapping is semantic of element / attribute names insignificant
  ⇒ we need a more suitable approach

Content

1. Overview of existing approaches
2. Proposed improvement
3. Experiments
4. Conclusion
Basic Ideas

- XML-to-relational mapping focuses on data structure
  - Complexity, data types, used constructs, ...
- Aim: similarity function $sim(f_x, f_y) \in [0,1]$
  - Schema fragments $f_x$ and $f_y$
  - $1 = \text{strong similarity}, 0 = \text{strong dissimilarity}$
- **Matcher** = evaluates similarity of a particular feature of $f_x$ and $f_y$
  - e.g. similarity of depths, number of elements / attributes, data types, ...
- **Composite similarity function** = aggregates results of matchers
  - Verified approach: weighted sum
Structural Aspects (1)

• Idea: Each matcher describes a structural aspect
  • Problem: How to state matchers?
• Idea: Exploitation of characteristics from statistical analyses of real-world data
  • Analyses: To analyze the data from various points of view
  • Our aim: To describe the data from various points of view
• Classification:
  • Root = characteristics of root node of schema fragment
    • e.g. type of content, element / attribute fan-out, …
  • Subtree = characteristics of the whole fragment
    • e.g. number of elements, depths, …
  • Level = characteristics of each level of fragment
    • e.g. number of attributes, minimum / maximum fan-outs, …

Structural Aspects (2)

• Transformation of values of matchers to [0,1]
  • Feature matchers – inequality of features
    • e.g. type of content

\[
m^{fea}_i(f_x, f_y) = \begin{cases} 
1 & fea_i(f_x) = fea_i(f_y) \\
0 & \text{otherwise}
\end{cases}
\]

• Single value matchers – difference of values
  • e.g. element fan-out

\[
m^{single}_j(f_x, f_y) = \frac{1}{|value_j(f_x) - value_j(f_y)| + 1}
\]
Structural Aspects (3)

- **Multi value matchers** – difference of sequences
  - e.g. allowed depths of fragments

\[
m_{j}^{\text{multi}}(f_x, f_y) = \frac{\sum_{k=1}^{m} \frac{1}{|s_j(f_x)[k] - s_j(f_y)[k]| + 1}}{m}
\]

- **Level matchers** – difference of values per levels
  - e.g. minimum and maximum fan-out per level

\[
m_{j}^{\text{lev}}(f_x, f_y) = \sum_{k=1}^{l} m_{j}^{\text{single/multi}}(f_x, f_y) \cdot \left(\frac{1}{2}\right)^k
\]
Tuning of Parameters (1)

- Problem: How to set the weights of composite similarity function?
  - Existing approaches: no care, average of values, machine learning
    - For semantic-based approaches suitable
    - For structure-based approaches not
- Idea: Exploitation of experience from the statistical analysis
  1. Use the same real-world data used in the analysis
  2. Prepare sample schema fragments with known representation in the data
  3. Compute occurrence of similar fragments in the data using the similarity function
  4. Tune the weights so that the results correspond to the results of the analysis
Tuning of Parameters (2)

- Theoretical view of the problem
  - Analysis:
    - $C_1, C_2, \ldots, C_K$ = categories of real-world schemes
    - $p_1, p_2, \ldots, p_P$ = sample schema patterns
    - $(M_{ij}^{rep})_{K \times P}$ = real-world representation of pattern $p_j$ in category $C_i$
  - Search algorithm:
    - Parameters $par_1, par_2, \ldots, par_R$, where $\forall i : par_i \in [0,1]$
    - With a setting of parameters returns calculated representation $rep_{ij}$ of pattern $p_j$ in category $C_i$
  - Aim: Optimal setting of parameters s.t. $\Delta$ is minimal
    $\Rightarrow$ A kind of constraints optimization problem (COP)

- Solution:
  - One of classical COP approaches
  - Genetic algorithms, simulated annealing, …

$$\Delta = \sum_{i=1}^{K} \sum_{j=1}^{P} |M_{ij}^{rep} - rep_{ij}|$$
Content

1. Overview of existing approaches
2. Proposed improvement
3. Experiments
4. Conclusion
Average and Tuned Weights

- \( R \) = manually determined matches, \( P \) = matches determined by algorithm
- \( I \) = true positives, \( F \) = false matches
- Precision = \( \frac{|I|}{|P|} \) = reliability of the function
- Recall = \( \frac{|I|}{|R|} \) = share of real matches that is found
- Overall = \( \frac{|I| - |F|}{|R|} \) = post-match effort
Content

1. Overview of existing approaches
2. Proposed improvement
3. Experiments
4. Conclusion
Conclusions and Future Work

• Our contributions:
  • A similarity function focusing on structural level
  • An approach for finding reasonable tuning of weights
    • A compromise between machine learning and straightforward setting
  • Both ideas can be simply extended to any appropriate similarity problem

• Future work:
  • Exploitation of semantic
    • Not a key aspect for XML-to-relational mapping, but can help in finding more reasonable mapping

Thank you