A Context-based Measure for Discovering Approximate Semantic Matching between Schema Elements

**Fabien Duchateau**, Zohra Bellahsène and Mathieu Roche

Laboratoire d’Informatique, de Robotique et de Microélectronique de Montpellier
Université Montpellier II, France

RCIS’07
Ouarzazate, Morocco
# Table of Content

1. Introduction and Motivations
   - Introduction
   - Contributions
   - A terminological example
   - A context example

2. Approxivect Approach
   - Some Notions
   - A 2-steps Matching Algorithm
   - Parameters
   - Experiments Results

3. Related Work

4. Conclusion and Future Work
1 Introduction and Motivations
   - Introduction
   - Contributions
   - A terminological example
   - A context example

2 Approxivect Approach
   - Some Notions
   - A 2-steps Matching Algorithm
   - Parameters
   - Experiments Results

3 Related Work

4 Conclusion and Future Work
Finding semantic correspondences between 2 schemas still a challenging issue

Semi automatic matchers available based on several approaches (combination of terminological measures, structural rules, ...)

### Motivations

Terminological measures are not sufficient, for example:
- mouse (computer device) and mouse (animal) ⇒ polysemy
- university and faculty ⇒ totally dissimilar labels

Structural measures have some drawbacks:
- propagating the benefit of irrelevant discovered matches to the neighbour nodes increases the discovering of more irrelevant matches
- not efficient with small schemas
Figure: Two schemas from the university domain.
Our approach: Approxivect

Based on the work of [1], Approxivect evaluates the similarity between two terms from different schema trees. It has the following properties:

- it is based on the combination of terminological measures (Levenhstein and n-grams) and structural measures (cosine measure applied to contexts)
- it is both automatic and not language-dependent
- it does not rely on dictionaries or ontologies
- it provides an acceptable matching quality
Figure: XML schemas relative to university.

- 3grams(Courses, GradCourses) = 0.2
- Lev(Courses, GradCourses) = 0.42

⇒ StringMatching(Courses, GradCourses) = 0.31
Figure: In the second schema, \textit{Courses} replaces \textit{GradCourses} due to StringMatching value.

- \textbf{StringMatching(Faculty, University)} = 0.002
- \textbf{Context(Faculty)} = Faculty, Courses, Professor
- \textbf{Context(University)} = University, Courses, Professor

$\Rightarrow$ \textbf{CosineMeasure(Context(Faculty), Context(University))} = 0.37
1 Introduction and Motivations
   • Introduction
   • Contributions
   • A terminological example
   • A context example

2 Approxivect Approach
   • Some Notions
   • A 2-steps Matching Algorithm
   • Parameters
   • Experiments Results

3 Related Work

4 Conclusion and Future Work
Context of node $n_c$

- represents the most important neighbour nodes $n_i$ for $n_c$
- each neighbour $n_i$ is assigned a weight depending on the relationship $n_c$

$$\omega(n_c, n_i) = 1 + \frac{K}{\Delta d + |\text{level}(n_c) - \text{level}(n_a)| + |\text{level}(n_i) - \text{level}(n_a)|}$$

String Matching is the average between

- Levenhstein distance
- 3-grams
Discovering semantic similarities:

- String Matching between 2 node labels
- if above a given threshold, replacement of one of the label by the other.

Cosine Measure using context:

- due to replacements, the contexts of two nodes can be very similar

**Similarity between two nodes**

It is the best value between String Matching and Cosine Measure.
NB_LEVELS restricts the context by limiting the number of levels
MIN_WEIGHT restricts the context by keeping only nodes with a weight above this threshold
REPLACE_THRESHOLD if the StringMatching between two node labels is above this replacement threshold, then one label is replaced by the other
κ represents the importance given to the context

Flexibility

These parameters allow more flexibility. Tuning them is required in some specific scenarios.
Some Notions
A 2-steps Matching Algorithm
Parameters
Experiments Results

Figure: Mappings discovered by an expert between the schemas.
### Some Notions

A 2-steps Matching Algorithm

#### Parameters

- Element from schema 1
- Element from schema 2
- Similarity value
- Relevance

<table>
<thead>
<tr>
<th>Element from schema 1</th>
<th>Element from schema 2</th>
<th>Similarity value</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Professor</td>
<td>1.0</td>
<td>+</td>
</tr>
<tr>
<td>CS Dept Australia</td>
<td>People</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>Courses</td>
<td>Grad Courses</td>
<td>0.41</td>
<td>+</td>
</tr>
<tr>
<td>CS Dept Australia</td>
<td>CS Dept U.S.</td>
<td>0.36</td>
<td>+</td>
</tr>
<tr>
<td>Courses</td>
<td>Undergrad Courses</td>
<td>0.28</td>
<td>+</td>
</tr>
<tr>
<td>Academic Staff</td>
<td>Faculty</td>
<td>0.25</td>
<td>+</td>
</tr>
<tr>
<td>Staff</td>
<td>People</td>
<td>0.23</td>
<td>+</td>
</tr>
<tr>
<td>Technical Staff</td>
<td>Staff</td>
<td>0.21</td>
<td>+</td>
</tr>
<tr>
<td>Senior Lecturer</td>
<td>Associate Professor</td>
<td>0.16</td>
<td>+</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Table:** Approxivect similarity ranking between the two schemas

<table>
<thead>
<tr>
<th>Element from schema 1</th>
<th>Element from schema 2</th>
<th>Similarity value</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Professor</td>
<td>0.53545463</td>
<td>+</td>
</tr>
<tr>
<td>Technical Staff</td>
<td>Staff</td>
<td>0.5300107</td>
<td>+</td>
</tr>
<tr>
<td>CS Dept Australia</td>
<td>CS Dept U.S.</td>
<td>0.52305263</td>
<td>+</td>
</tr>
<tr>
<td>Courses</td>
<td>Grad Courses</td>
<td>0.5041725</td>
<td>+</td>
</tr>
<tr>
<td>Courses</td>
<td>Undergrad Courses</td>
<td>0.5041725</td>
<td>+</td>
</tr>
</tbody>
</table>

**Table:** COMA++ discovered mappings between the two schemas
### Table: Results of COMA++ and Approxivect on the XML schemas

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMA++</td>
<td>1</td>
<td>0.56</td>
<td>0.72</td>
</tr>
<tr>
<td>Approxivect</td>
<td>0.62</td>
<td>0.89</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Note that Approxivect parameters are set to default. An optimal configuration enables to obtain a 0.82 F-measure.
1 Introduction and Motivations
   • Introduction
   • Contributions
   • A terminological example
   • A context example

2 Approxivect Approach
   • Some Notions
   • A 2-steps Matching Algorithm
   • Parameters
   • Experiments Results

3 Related Work

4 Conclusion and Future Work
COMA++ [2]

- combination of many terminological measures and a user-defined synonym table
- a matrix is built for each couple of elements and for each measure
- a strategy is applied to select the mappings
- mappings are modified and/or validated by the user

Similarity Flooding [3]

- a simple string matching algorithm to provide initial matchings
- structural rules and propagation to refine the matchings
- mappings are modified and/or validated by the user
Introduction and Motivations
  - Introduction
  - Contributions
  - A terminological example
  - A context example

Approxivect Approach
  - Some Notions
  - A 2-steps Matching Algorithm
  - Parameters
  - Experiments Results

Related Work

Conclusion and Future Work
An automatic schema matching approach

- based on the combination of terminological and structural measures
- with an acceptable quality of matching
- flexible thanks to the parameters

However

- tuning is not automatic, but some tools could handle this step (eTuner)
- more heterogeneity in the experiments

Ongoing work

- performance aspect
