Ontology matching evaluation using GIS services

Lorenzino Vaccari, Juan Pane,
DISI Department
University of Trento, Italy
email: vaccari/pane@disi.unitn.it
Outline

• Introduction
• Structure Preserving Semantic Matching
• Evaluation: two test cases
  1. Evolution scenario: *syntactic and semantic alterations*
  2. Classification comparison
• Performance evaluation
• Conclusions and future work
Introduction

How do we use matching?

1. **(Web Service Discovery)** to allow service providers to determine how similar their own service descriptions are to those required by service requestors *(service invocations)*.

2. **(Web Service Invocation)** to allow service providers to understand how they may satisfy the requirements of service requestors. This is done through building up a map between each element of a service description to each element of a service invocation.

Ontology matching approach:

- **Structure Preserving Semantic Matching (SPSM)**. The semantic matching *preserves* some structural properties (e.g., functions are matched to functions and variables are matched to variables)
SPSM

1. The similarity (Treesim) between service invocation signature (tree \( T_1 \)) and service description signature (tree \( T_2 \)) is 0.64

2. 

\begin{verbatim}
getMap
--- Version
--- Layers
--- Width
--- Height
--- Format
--- Xmin_BB
--- Ymin_BB
--- Xmax_BB
--- Ymax_BB

getMap
--- Dimension
--- Width
--- Height
--- Edition
--- Layers
--- DataFormat
--- Request
--- Xmin
--- Ymin
--- Xmax
--- Ymax
\end{verbatim}
Test cases

**TC1: Syntactic and semantic robustness test:**
- Using alteration operations on Web service operation signatures
- The probability, assigned to each alteration operation, has been changed from the lower value (0.1) to the maximum value (0.9)

**TC2: Classification robustness test:**
- Comparison between a manual classification and the one computed by SPSM
TC1: Evaluation dataset

- **80 trees** were built out of the Geographic Web services
- **4 alteration operations + 1 combination:**
  - Meaning and syntactic alterations
- **20 alterations** for each tree, for each alteration operation and for each probability
  - total matching tasks for each alteration operation: 14,400
  - total matching tasks (including 10 statistical repetitions): ca. 700,000
TC1: alteration operations

Original signature:
find_Address_By_Point(point, address_Finder_Options, part)

1. Replace a node name with an unrelated one (Brown corpus):
   point $\rightarrow$ cable

2. Add or remove a label in a node name (Brown corpus):
   find_Address_By_Point $\rightarrow$ find_By_Point

3. Alter syntactically a label (add, delete and change characters):
   find_Address_By_Point $\rightarrow$ finm_Address_By_Poioat

4. Replace a label in a node name with a related one
   (synonyms, hyponyms, hypernyms from Moby and WordNet 3.0)
   address_Finder_Options $\rightarrow$ location_Finder_Options

5. Combination of 3. and 4.
   address_Finder_Options $\rightarrow$ Ifctin_Finder_Options
TC1: evaluation methodology

• Initially, to each pair of signatures is assigned a starting similarity.
  – $ExpScore \leftarrow 1$

• Alteration operations application:
  – $ExpScore$ reduction

• Recall, Precision and F-measure values can be computed. Ingredients:
  – $ExpScore$
  – $TreeSim$ (SPSM)
  – Variable acceptance (cut-off) threshold

• Results: average on 10 repetitions
  – average standard deviation: 1.3%
TC1: evaluation results, example

- **Replace a node name with an unrelated node name**
TC1: SPSM vs baseline (edit distance)

- **Syntactic alterations** (alteration probability = 0.6)
- **Meaning alterations** (alteration probability = 0.6)

**F-measure of SPSM matcher:**
- equivalent for syntactic alterations,
  - +20% for meaning alterations

**SPSM matcher:** “best of both worlds”
TC2: evaluation dataset and setup

• **Selected set** (50) of GIS Web service operations.
• **Manual classification:**
  • WSDL description
  • Eliminate some general operations
  • Refine classification by regrouping operations
## TC2: evaluation methodology

### Manual classification, example

<table>
<thead>
<tr>
<th>Operation 1</th>
<th>Operation 2</th>
<th>Operation 3</th>
<th>Operation 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Operation 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Operation 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Operation 4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### SPSM classification, example

<table>
<thead>
<tr>
<th>Operation 1</th>
<th>Operation 2</th>
<th>Operation 3</th>
<th>Operation 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation 1</td>
<td>1</td>
<td>0.76</td>
<td>0.22</td>
</tr>
<tr>
<td>Operation 2</td>
<td>0.76</td>
<td>1</td>
<td>0.57</td>
</tr>
<tr>
<td>Operation 3</td>
<td>0.22</td>
<td>0.57</td>
<td>1</td>
</tr>
<tr>
<td>Operation 4</td>
<td>0.52</td>
<td>0.54</td>
<td>0.12</td>
</tr>
</tbody>
</table>

### Cut-off Threshold = 0.5

<table>
<thead>
<tr>
<th>Relevant</th>
<th>Retrieved</th>
<th>True Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

Recall = 0.8; Precision = 0.67; F-measure = 0.73
TC2: evaluation results

- Best F-measure: 52%
Performance evaluation

- Setup: standard laptop Intel Centrino Core Duo CPU-2Ghz, 2GB RAM, Windows Vista O.S., no applications running but a single matching system.
- Average numbers of the parameters of the WSDL operations: 4
- Efficiency:
  - Execution Time per matching task: 43 ms
  - Quantity of main memory during matching tasks: less than 2.3Mb (than the standby level)
Conclusions and Future work

• Results:
  – Test case 1. Evaluation results: +20% in comparison to the baseline
  – Test case 2. SPSM supports classification of real world web services with best F-measure around 52%

• Applicability:
  • It can be applied to web service discovery and invocation
  • It is robust and it can be used at run-time

• Future work:
  – Use domain specific (GIS) and/or multilingual thesauri, e.g. Gemet, Agrovoc and Eurovoc for semantic matching
  – Compare results in “live” environments with community of users, e.g. e-Health, e-Response and GIS
  – Geo-data similarity evaluation, e.g. INSPIRE themes and GIS schema specifications
Thank you for your attention!

QUESTIONS?

This work has been supported by the EU project OpenKnowledge

www.openk.org

(Matching: WP3)