S-Match

7 years of research and exploitation
S-Match: an algorithm and an implementation of semantic matching

Fausto Giunchiglia, Pavel Shvaiko, Mikalai Yatskevich
Dept. of Informatics and Communication Technology
University of Trento
38050 Povo, Trento, Italy
{fausto, pavel, parshive}@dit.unitn.it

Abstract. We think of S-Match as an operator which takes two graph-like structures (e.g., conceptual hierarchies or ontologies) and produces mappings between the nodes of the two graphs that correspond semantically to each other. S-Match is a critical operator in many well-known application domains, such as ontology integration, data warehouses, and XML message mapping. More recently, new application domains have emerged, such as catalog matching, where the match operator is used to map entries of catalogs among business partners, or web service coordination, where match is used to identify dependencies among data sources.

We concentrate on semantic matching, an introduced in [8], based on the idea and system described in [2]. The key intuition behind semantic matching is that we should calculate mappings by comparing the semantic relationships holding between the concepts. Semantic relationships are a more abstract and more expressive way of expressing equivalence, one can be more general than the other, and so on. We classify all previous approaches under the heading of syntactic matching. These approaches, though implicitly or explicitly exploiting the semantic information codified in graphs, differ substantially from our approach in this: instead of comparing semantic relations between nodes, they compute semantic “similarity” coefficients between labels, in the [0,1] range. Some examples of previous solutions are [12], [1] [15], [10], [3] [10]; see [1] for a nice-depth discussion about syntactic and semantic matching.

In this paper we propose and analyze in detail an algorithm and a system implementing semantic matching. Our approach is based on two key notions, the notion of
The Team

S-Match
Fausto Giunchiglia
Pavel Shvaiko
Mikalai Yatskevich
Aliaksandr Autayeu

Minimal Mappings
Fausto Giunchiglia
Vincenzo Maltese
Aliaksandr Autayeu

Structure Preserving
Semantic Matching
Fausto Giunchiglia
Juan Pane
Lorenzino Vaccari
Gaia Trecarichi
Mikalai Yatskevich

Background Knowledge
Datasets
Fausto Giunchiglia
Vincenzo Maltese
Feroz Farazi
Biswanath Dutta

Fausto Giunchiglia
Pavel Shvaiko
Mikalai Yatskevich
Aliaksandr Autayeu

Gaia Trecarichi
Juan Pane
Lorenzo Vaccari
Feroz Farazi
Biswanath Dutta
Vincenzo Maltese
Overview

- Introduction to S-Match
- Lightweight Ontologies
- Matching Tools
  - S-Match
  - Structure Preserving Semantic Matching (SPSM)
  - MinSMatch for minimal mappings
- Evaluations
- Enhancements: NLP, BK
- Open Source Framework
- Exploitation
- Future activities
Living with heterogeneity [KER-03]

- The semantic web will be: huge, dynamic and heterogeneous. These are not bugs, these are features

- We must learn to live with them and master them

- Often information resources expressed in different ways must be reconciled before being used. Mismatch between formalized knowledge can occur when:
  - different languages are used
  - different terminologies are used
  - different modeling is used
On reducing heterogeneity [KER-03]

Reconciliation can be performed in 2 steps:

(i) match,

thereby determine an alignment,

(ii) generate

a processor (for transformation, etc.)
2004: what made the difference?

- About 30+ matching systems existed in 2004
  - Cupid, COMA, Rondo, NOM, OLA, Prompt, Anchor-Prompt, CtxMatch, …
  - now 100+ systems exist

- [0..1] vs. \{ =, < , > , ⊥ \}
  - Most systems were computing and aggregating various similarity measures in [0 1] to produce alignments
  - We computed logical relations: equivalence, subsumption, …

- Heuristics vs. soundness and completeness
  - Most systems were using matching heuristics that sometimes worked well, sometimes not so well. We followed this path as well, but…
  - One step of the matching process was sound and complete
What is Semantic Matching [KER-03]

- An operation that identifies semantically similar nodes in two graph-like structures

- Applications: catalog integration, peer to peer information sharing, resource discovery, query answering, …
The Key Idea [KER-03, ESWS-04]

- Take as input two graph-like structures, e.g., ontologies
- Return as output logic relations, e.g., equivalence, subsumption, which are supposed to hold between the nodes of the graphs
- Entities of the input ontologies are translated into propositional formulas which explicitly express the concept descriptions as encoded in the ontology structure and in external resources, such as WordNet
- Translation of the matching problem into a propositional validity problem
- Propositional validity problem, efficiently resolved using sound and complete propositional satisfiability (SAT) solvers
S-Match Algo Key Steps [ESWS-04]

- Given two trees (lightweight ontologies) T1 and T2:

1. For all labels in T1 and T2 compute concepts at labels (analysis of labels in isolation; from natural language to propositional logic)
2. For all nodes in T1 and T2 compute concepts at nodes (take into account structure of the trees)
3. For all pairs of labels in T1 and T2 compute relations between atomic concepts at labels (build Theory)
4. For all pairs of nodes in T1 and T2 compute relations between concepts at nodes (run SAT)

- Steps 1, 2: preprocessing phase (once for all)
- Steps 3, 4: matching phase (run-time)
Lightweight Ontologies [JODS-05]

- Lightweight ontologies are tree structures where concepts at nodes are connected with subsumption in DL.

- Many of the schemas in the world can be translated into lightweight ontologies:
  - User classifications (file systems, email folder structures)
  - Web directories and business catalogues
  - Library classifications (thesauri, subject headings)

- **With the translation:**
  - Node labels are formulas in propositional Description Logic (DL)
  - Concepts are taken from WordNet senses (or other dictionaries)
  - Tree structures: each node formula is subsumed by parent node formula
Lightweight Ontologies (cont)

JOURNALS
journals\#1

DEVELOPMENT AND PROGRAMMING LANGUAGES
(development\#1 \sqcup \text{programming}\#2)
\sqcap \text{languages}\#3 \sqcap \text{journals}\#1

JAVA
(development\#1 \sqcup \text{programming}\#2)
\sqcap \text{languages}\#3 \sqcap \text{journals}\#1
\sqcap \text{Java}\#3

PROGRAMMING AND DEVELOPMENT
\text{programming}\#2 \sqcup \text{development}\#1

LANGUAGES
\text{languages}\#3 \sqcap (\text{programming}\#2 \sqcup \text{development}\#1)

JAVA
\text{Java}\#3 \sqcap \text{languages}\#3 \sqcap (\text{programming}\#2 \sqcup \text{development}\#1)

MAGAZINES
\text{Magazines}\#1 \sqcap \text{Java}\#3 \sqcap \text{languages}\#3
\sqcap (\text{programming}\#2 \sqcup \text{development}\#1)
Matching Tools

- S-Match: the basic semantic matching tool
  - It returns the set of semantic correspondences between two lightweight ontologies
  - Output: ⊥, ⊳, ⊲, ≡

- SPSM: Structure Preserving Semantic Matching
  - Only one correspondence per node is returned
  - It matches leaf nodes to leaf nodes and internal nodes to internal nodes
  - Used to compare function definitions

- MinSMatch: to compute minimal mappings
  - It returns the minimal set of semantic correspondences between two lightweight ontologies. It always exists and it is unique
  - It computes the set of maximum size (containing the maximum number of minimal and redundant links) from the propagation of the links in the minimal set

- S-Match GUI
S-Match [ESWS-04]

- An alignment is a set of mapping elements <source, target, R>
  - R ∈ { ‘⊥’, ‘≡’, ‘⊑’, ‘⊒’} partially ordered
  - For each pair of nodes a call to a SAT solver verifies if a given semantic relation holds between the two, given the available background knowledge
  - Visualization and usability problems (e.g. validation and maintenance)
SPSM [ODBASE-08a]

- SPSM: Structure Preserving Semantic Matching

  - Example with two web services:
    - Get_Wine(Region, Country, Color, Price, Number_of_bottles)
    - Get_Wine(Region(Country, Area), Colour, Cost, Year, Quantity)
    - SPSM (T1,T2) = 0.62 + set of mapping elements

  - Uses abstraction operations to preserve structures, namely it computes one-to-one correspondence, such that:
    - Functions are matched to functions
    - Variables are matched to variables

  - Outputs a global similarity measure and a set mapping elements.

- Node matching is done with S-Match

- A global similarity measure is computed using Tree edit distance
Based on a set of redundancy patterns the minimal mapping is that minimal subset of correspondences such as all the others can be efficiently computed from them.

The minimal mapping always exists and it is unique.
S-Match GUI [SWJ-10]

- Traditional visualization: crowded already with only 34x39 nodes
S-Match GUI [SWJ-10]

- New GUI
  - node-links
  - ellipsis
  - hints
  - path-to-root
  - links table
  - editing
  - synchronized navigation
### MinSMatch Evaluation [ODBASE-10]

**Mapping sizes and percentage of reduction on standard datasets**

<table>
<thead>
<tr>
<th>Datasets (nodes)</th>
<th>Mapping of maximum size</th>
<th>Minimal Mapping size</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Cornell/Washington (34/39)</td>
<td>223</td>
<td>36</td>
<td>83.86</td>
</tr>
<tr>
<td>#2 Topia/Icon (542/999)</td>
<td>5491</td>
<td>243</td>
<td>95.57</td>
</tr>
<tr>
<td>#3 Web dir. Source/Target (2857/6628)</td>
<td>282648</td>
<td>30956</td>
<td>89.05</td>
</tr>
<tr>
<td>#4 EClass/UNSPSC (3358/5293)</td>
<td>39818</td>
<td>12754</td>
<td>67.97</td>
</tr>
</tbody>
</table>

### Reduction in run time and calls to SAT

<table>
<thead>
<tr>
<th>#</th>
<th>Run Time (ms)</th>
<th>Calls to logical reasoner (SAT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S-Match</td>
<td>MinSMatch</td>
</tr>
<tr>
<td>1</td>
<td>472</td>
<td>397</td>
</tr>
<tr>
<td>2</td>
<td>141040</td>
<td>67125</td>
</tr>
<tr>
<td>3</td>
<td>3593058</td>
<td>1847252</td>
</tr>
<tr>
<td>4</td>
<td>6440952</td>
<td>2642064</td>
</tr>
</tbody>
</table>
MinSMatch Evaluation [ODBASE-10]

Mapping sizes and percentage of reduction on NALT and LCSH

<table>
<thead>
<tr>
<th>Id</th>
<th>Source</th>
<th>Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>NALT</td>
<td>Chemistry and Physics</td>
</tr>
<tr>
<td>B</td>
<td>NALT</td>
<td>Natural Resources, Earth and Environmental Sciences</td>
</tr>
<tr>
<td>C</td>
<td>LCSH</td>
<td>Chemical Elements</td>
</tr>
<tr>
<td>D</td>
<td>LCSH</td>
<td>Chemicals</td>
</tr>
<tr>
<td>E</td>
<td>LCSH</td>
<td>Management</td>
</tr>
<tr>
<td>F</td>
<td>LCSH</td>
<td>Natural resources</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branches</th>
<th>Mapping of maximum size</th>
<th>Minimal mapping size</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A vs. C</td>
<td>17716</td>
<td>7541</td>
<td>57.43</td>
</tr>
<tr>
<td>A vs. D</td>
<td>139121</td>
<td>994</td>
<td>99.29</td>
</tr>
<tr>
<td>A vs. E</td>
<td>9579</td>
<td>1254</td>
<td>86.91</td>
</tr>
<tr>
<td>B vs. F</td>
<td>27191</td>
<td>1232</td>
<td>95.47</td>
</tr>
</tbody>
</table>
Improved NLP [ISWC-07, ECDL-10]

- Classifications, database schemas, APIs...
- Natural Language Metadata: labels, very short pieces of text
  - short context to no context
  - special syntax tools
  - biased toward nouns, distribution of parts of speech
- Improved NLP: manual annotation + language analysis
  - tokenization
  - parts of speech tagging
  - lightweight parsing: simple NP-based grammar
- +18% in translation accuracy
Improved BK [ECAI-06, ISWC-10]

BK: Background Knowledge

- **WordNet**
  - http://wordnet.princeton.edu
  - general, small, single language
  - ~120K concepts, covers daily language

- **GeoWordNet**
  - http://geowordnet.semanticmatching.org/
  - specific, huge, several languages
  - ~3.6M+ entities, 7.2M+ relations, world places

- **Entitypedia**
  - http://entitypedia.org/
  - general, huge, multilingual,
  - covers world entities and domains, coming soon…
Open Source Framework [SM, SWJ-10]

- http://semanticmatching.org/ since March 2010

- SF.net community
- Source Code
- Documentation
- Data sets

- LGPL
- CC-BY
- almost 2000 dls
Exploitation

Semantic Geo-Catalog (SGC)
S-Match is used to match a user query to a faceted ontology in the geo-spatial domain

Experiments in the agriculture domain
S-Match to match AGROVOC with CABI

Interconcept
MinSMatch to match Knowledge Organization Systems in digital libraries

Open Knowledge
SPSM to match web services

And others …
The query expansion component integrated with the geo-catalog
- The local dataset of the Province of Trento has been used to construct the faceted ontology and integrated with GeoWordNet
Semantic Matching: Theory and Practice

by Fausto Giunchiglia and Aliaksandr Autayeu

end 2011 - beginning 2012

- Fundamentals
  - Introduction to Semantic Matching
  - Lightweight Ontologies
  - Basic Algorithm
  - Structure Preserving Semantic Matching
  - Minimal Semantic Matching
  - Non-Standard Uses of Matching

- The Framework
  - Introduction to the S-Match
  - Input: Everything is a Tree
  - Processing Natural Language Metadata
  - Background Knowledge

- ... The Framework
  - ...
  - Background Knowledge
  - Element-level Matching
  - Structure-level Matching
  - Advanced Matching
  - Output: Semantic Mappings
  - Framework Extensions

- Datasets and Evaluation
  - Evaluation Issues and Methodology
  - Datasets
  - Evaluating Conversion into Lightweight Ontologies
  - Evaluating Matching Techniques
Other Relevant Initiatives

- **OAEI: Ontology Alignment Evaluation Initiative**
  - since 2004, supported by
    - Pavel Shvaiko, Mikalai Yatskevich, Juan Pane

- **Ontology Matching Workshop at ISWC**
  - since 2006, supported by
    - Pavel Shvaiko, Fausto Giunchiglia

- **Book on Ontology matching [OMB-07]**
  - In 2007, by Pavel Shvaiko and others
References

- [ECAI-06] F. Giunchiglia, P. Shvaiko, M. Yatskevich: Discovering Missing Background Knowledge in Ontology Matching. ECAI 2006
Thank you for your time and interest!

Questions?

http://semanticmatching.org/