Centrum voor Wiskunde en Informatica

Aligner automatiquement des ontologies avec 

Raphaël Troncy

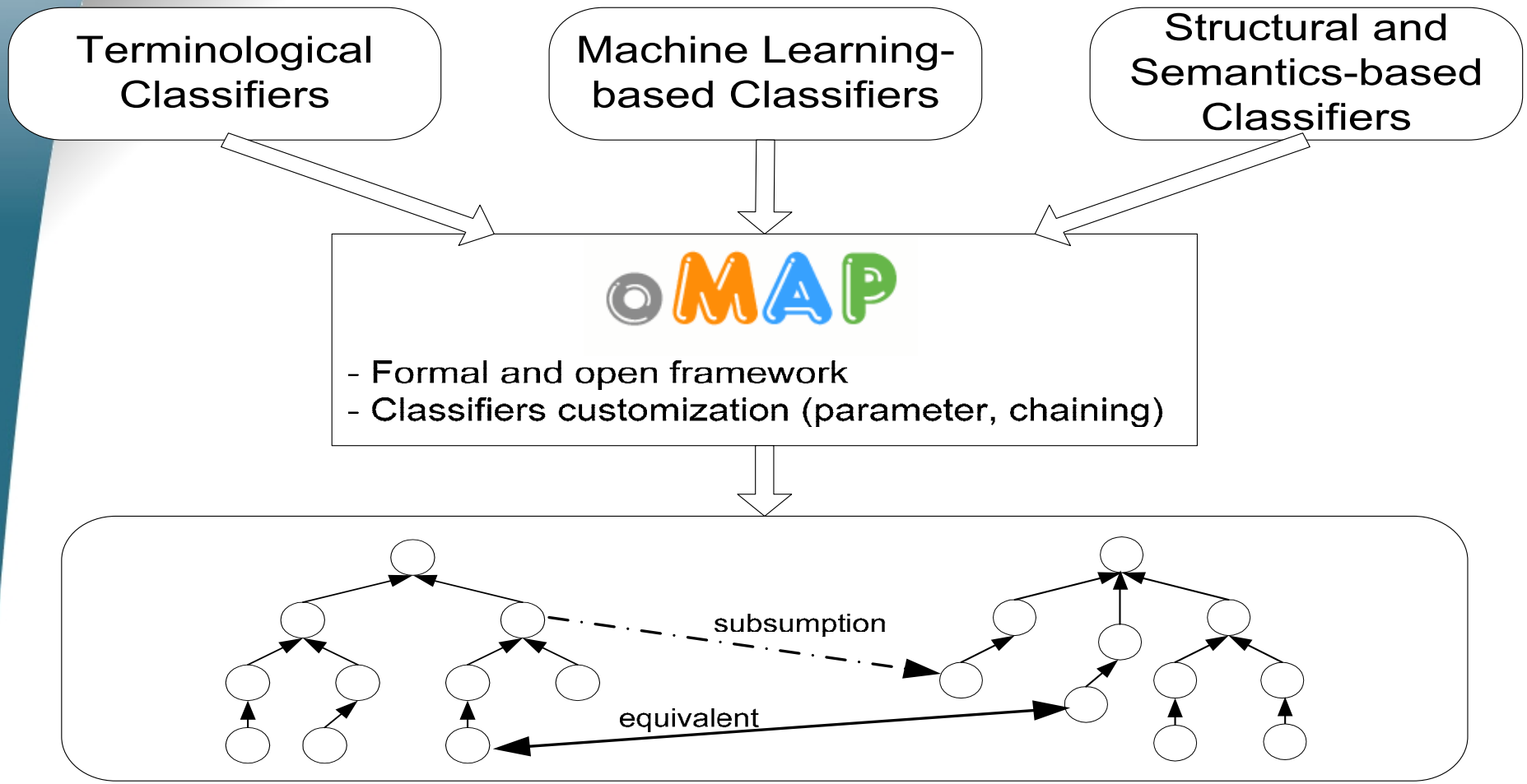
Tuesday 23rd of January, 2007

Raphael.Troncy@cwi.nl

Motivation

- Various SW repositories, using different vocabularies, distributed on the web
- Already large amounts of data out there
  - Swoogle hits 10^7 – 10^9 unique Semantic Web documents (16/11/2006)
- Problem:
  - How to search and retrieve information in such an environment?
  - How to map the various vocabularies used ?

oMAP: Ontology Alignment Tool



oMAP: A Formal Framework

- Sources of inspiration:
  - Formal work in data exchange [Fagin *et al.*, 2003]
  - GLUE: combining several specialized components for finding the best set of mappings [Doan *et al.*, 2003]
- Notation:
  - A mapping is a triple: $M = (T, S, \sum)$
    - S and T are the *source* and *target* ontologies
    - S_i is an OWL entity (class, datatype property, object property) of the ontology
    - $\sum$ is a set of mapping rules: $\alpha_{ij}\ T_j \leftarrow S_i$

oMAP: Combining Classifiers

- Weight of a mapping rule:
  - $\alpha_{ij} = w(S_i, T_j, \sum)$
- Using different classifiers:
  - $w(S_i, T_j, CL_k)$ is the classifier's approximation of the rule $T_j \leftarrow S_i$
- Combining the approximations:
  - Use of a priority list: $CL_1 \prec CL_2 \prec \ldots \prec CL_n$
  - Weighted average of the classifiers prediction

Terminological Classifiers

- Same entity names (or URI)

$$w(S_i, T_j, CL_N) = \begin{cases} 1 & \text{if } S_i, T_j \text{ have same name,} \\ 0 & \text{otherwise} \end{cases}$$

- Same entity name stems

$$w(S_i, T_j, CL_S) = \begin{cases} 1 & \text{if } S_i, T_j \text{ have same stem,} \\ 0 & \text{otherwise} \end{cases}$$

Terminological Classifiers

- String distance name

$$w(S_i, T_j, CL_{LD}) = \frac{dist_{Levenshtein}(S_i, T_j)}{\max(length(S_i), length(T_j))}$$

- Iterative substring matching

$$w(S_i, T_j, CL_{IS}) = Comm(S_i, T_j) - Diff(S_i, T_j) + winkler(S_i, T_j)$$

  - See [Stoilos *et al.*, ISWC'05]

Terminological Classifiers

- WordNet distance name

$$w(S_i, T_j, CL_{WN}) = \begin{cases} 1 & \text{if } S_i, T_j \text{ are synonyms,} \\ \max\left(sim, \dfrac{2 * lcs}{length(S_i) + length(T_j)} \right) & \text{otherwise} \end{cases}$$

  - *lcs* is the longest common substring between S_i and T_j
  - *sim* = $\dfrac{| synonym(S_i) | \bigcap | synonym(T_j) |}{| synonym(S_i) | \bigcup | synonym(T_j) |}$

Machine Learning-Based Classifiers

- Collecting bag of words:
  - *label* for the named individuals
  - *data value* for the datatype properties
  - *type* for the anonymous individuals and the range of object properties
  - ...
- Recursion on the OWL definition:
  - *depth* parameter
- Use statistical methods on the collected bag of words

Machine Learning-Based Classifiers

- Example

  Individual (x_1 type (Workshop)
  value (label "DECOR") value (location x_2))
  Individual (x_2 type (Address)
  value (city "Namur") value (country "Belgium"))
  u1 = (" DECOR ", "Address")
  u2 = ("Address", "Namur", "Belgium")

- Naïve Bayes text classifier

$$w(S_i, T_j, CL_{NB}) = \Pr(S_i) \cdot \sum_{(x,u) \in T_j} \prod_{m \in u} \Pr(m | S_i)$$

- kNN text classifier

Structural and Semantics-Based Classifier

- $\sum$ is a set of mapping rules: $\alpha_{ij}\ T_j \leftarrow S_i$
- $\sum$ sets are computed by taking the OWL definition of the entities to align
  - recursively in the OWL structure
  - ... without looping thanks to cycles detection

Structural and Semantics-Based Classifier

- If S_i and T_j are property names:

$$w(S_i, T_j, \Sigma) = \begin{cases} 0 & \text{if } T_j \leftarrow S_i \notin \Sigma \\ w'(S_i, T_j, \Sigma) & \text{otherwise} \end{cases}$$

- If S_i and T_j are concept names[1]:

$$w(S_i, T_j, \Sigma) = \begin{cases} 0 & \text{if } T_j \leftarrow S_i \notin \Sigma \\ w'(S_i, T_j, \Sigma) & \text{if } |\mathrm{D}| = 0 \text{ and } T_j \leftarrow S_i \in \Sigma \\ \frac{1}{(|\mathrm{Set}| + 1)} \cdot \left(w'(S_i, T_j, \Sigma) + \max_{set} \left(\sum_{(Ci,Dj) \in set} w(C_i, D_j, \Sigma) \right) \right) & \text{otherwise} \end{cases}$$

[1] **Where $D = D(S_i) * D(T_j)$; $D(S_i)$ represents the set of concepts directly parent of S_i**

Structural and Semantics-Based Classifier

- Let $C_S=(QR.C)$ and $D_T=(Q'R'.D)$, then[1]:

$$w(C_S, D_T, \Sigma) = w_Q(Q, Q') \cdot w(R, R', \Sigma) \cdot w(C, D, \Sigma)$$

- Let $C_S=(op\ C_1 \ldots C_m)$ and $D_T=(op'\ D_1 \ldots D_m)$, then[2]:

$$w(C_S, D_T, \Sigma) = w_{op}(op, op') \cdot \frac{\max_{set}\left(\sum_{(Ci,Dj)\in set} w(C_i, D_j, \Sigma)\right)}{\min(m, n)}$$

[1] Where Q,Q' are quantifiers, R,R' are property names and C,D concept expressions

[2] Where op, op' are concept constructors and $n,m \geq 1$

Evaluation

- OAEI Contests (2004, 2005, 2006): http://oaei.ontologymatching.org/
  - Systematic benchmark tests on bibliographic data
    - Tests 2xx: aligning an ontology with variations of itself where each OWL constructs are discarded or modified one per one
    - Tests 3xx: four real bibliographic ontologies
  - Web categories alignment
  - http://oaei.ontologymatching.org/2005/results/

Benchmark Tests (2005)

Precision	
dublin20	0.92
Falcon	0.91
FOAM	0.90
oMAP	0.85
CMS	0.81
OLA	0.80
ctxMatch	0.72
edna	0.45

Recall	
Falcon	0.89
OLA	0.74
dublin20	0.72
FOAM	0.69
oMAP	0.68
edna	0.61
ctxMatch	0.20
CMS	0.18

- oMAP:
  - 4th with the global F-Measure
  - 1st on 3xx tests (*real* ontologies to align)

Aligning Web Categories (2005)

- Aligning Google, Loksmart and Yahoo web categories [Avesani *et al.*, ISWC'05]
- Blind tests: only recall results are available

ctxMatch	FOAM	CMS	Dublin20	Falcon	OLA	oMAP
9.4%	11.9%	14.1%	26.5%	31.2%	32.0%	34.4%

Conclusion

- oMAP: a formal framework for aligning automatically OWL ontologies
- Combining several specific classifiers
  - Terminological classifiers
  - Machine learning-based classifiers
  - Structural and semantics-based classifier

Future Work

- oMAP
  - Using additional classifiers:
    - KL-distance, other resources, background K, etc.
    - Straightforward theoretically but practically difficult!
  - Finding *complex* alignment
    - *name* = *firstName* + *lastName*
  - OWL and rule-based languages:
    - Take into account this additional expressivity
  - Other KR languages: e.g. SKOS

http://www.cwi.nl/~troncy/oMAP/

Any questions ?

Structural and Semantics-Based Classifier

- Possible values for w_{op} and w_Q *weights*

w_{op}

	$\sqcap$	$\sqcup$	$\neg$
$\sqcap$	1	1/4	0
$\sqcup$		1	0
$\neg$			1

w_Q

	$\exists$	$\forall$
$\exists$	1	1/4
$\forall$		1

	$\leq n$	$\geq n$
$\leq m$	1	1/3
$\geq m$		1