Comparison of schema matching evaluations

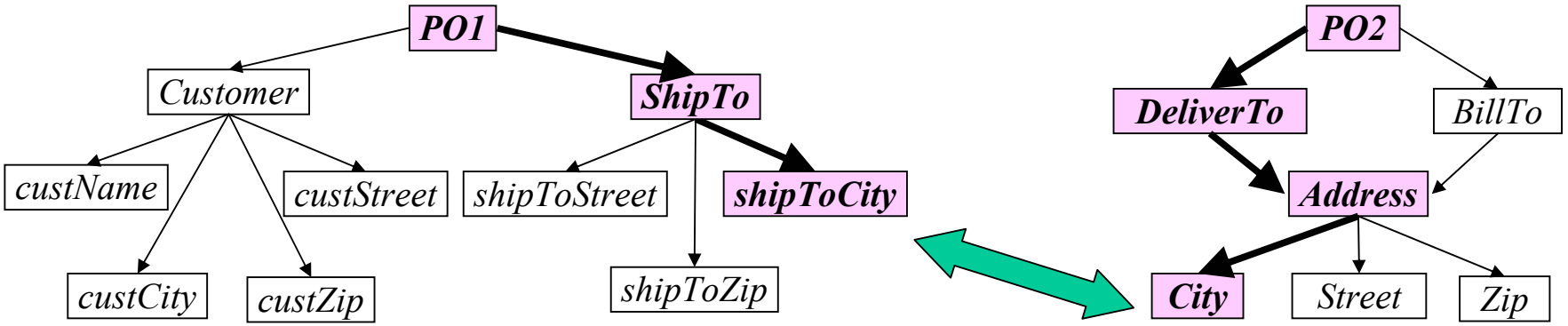
Hong-Hai Do, Sergey Melnik, Erhard Rahm
University of Leipzig, Germany
dbs.uni-leipzig.de

Content

- Motivation
- Comparison criteria
- Comparison: 8 Evaluations
- Conclusions

Motivation

- Schema matching: Finding semantic correspondences, i.e. matches, between two schemas



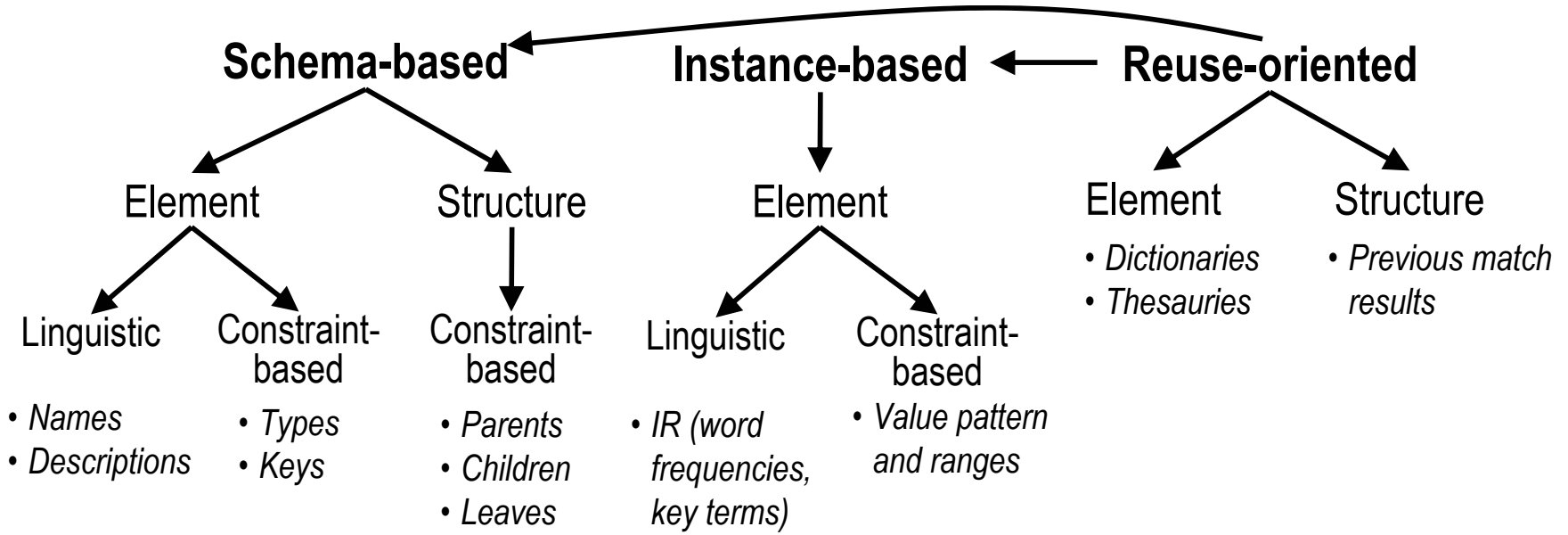
PO1.ShipTo.shipToCity ↔ PO2.DeliverTo.Address.City

- Crucial step in many applications
  - Schema and data integration: mediators, data warehouses
  - E-Business: XML message mapping
  - ...
- Currently manual, tedious, time-consuming
  - Need for (semi-)automatic approaches

Automatic Match Approaches

Survey [Rahm, Bernstein - VLDBJ01]

- Individual approaches



- Combined approaches
  - *Hybrid*: fixed combination of multiple match criteria
  - *Composite*: combination of the results of independently executed matchers

Existing Systems and Evaluations

- Existing automatic schema matching systems
  - Autoplex and Automatch (George Mason Univ.)
  - COMA (Univ. of Leipzig)
  - Cupid (Univ. of Washington)
  - LSD and GLUE (Univ. of Washington)
  - Similarity Flooding (Stanford Univ./Univ. of Leipzig)
  - SemInt (Northwestern Univ.)

  - Not evaluated/no sufficient info: Clio, Delta, DIKE, MOMIS-ARTEMIS, SKAT, TranScm

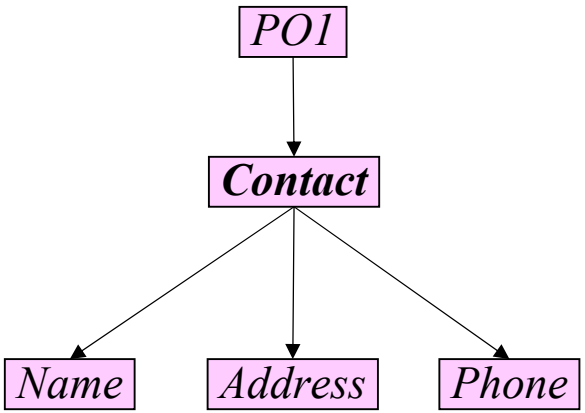
- Questions
  - Quality of the evaluation?
  - Quality of the system/approach?
  - The best system/approach? For what kind of problems?

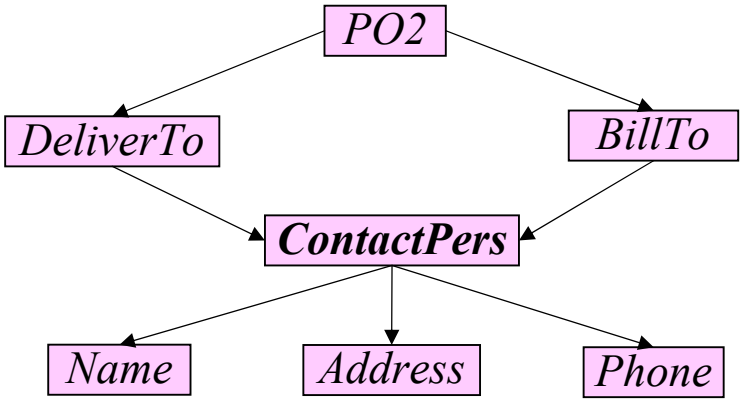
Comparison Criteria

- Areas of criteria
  - Input
  - Output
  - Quality measures
  - Test methodology

- Criteria: Input
  - Test schemas and match tasks
    - Schema information: language, size, etc.
    - Test problems: #schemas, #match tasks, etc.
  - Auxiliary information
    - Synonym tables, thesauries, constraints, etc.

Criteria: Output



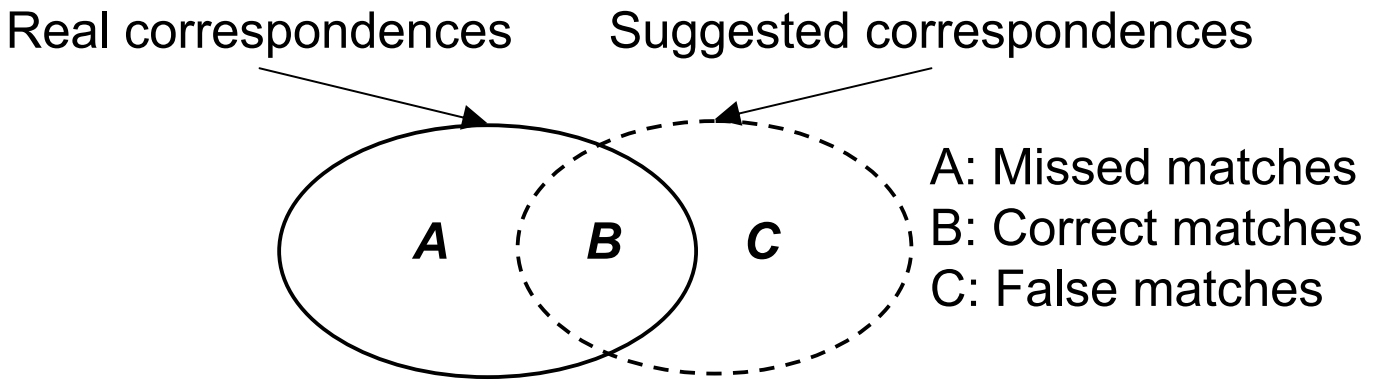


- Element representation: correspondences between *nodes* vs. between *paths*
  - [*Contact* ↔ *ContactPers*] vs. [*PO1.Contact* ↔ *PO2.DeliverTo.ContactPers*]
- Correspondence cardinality: *1:1* vs. *set-oriented*
  - [*PO1.Contact* ↔ *PO2.DeliverTo.ContactPers*],
    [*PO1.Contact* ↔ *PO2.BillTo.ContactPers*]

  vs.
  - [*PO1.Contact* ↔ *PO2.DeliverTo.ContactPers*, *PO2.BillTo.ContactPers*]

Criteria: Quality Measures

- Comparison of *automatically* with *manually* (i.e. real) derived correspondences



- Common quality measures (Information retrieval)

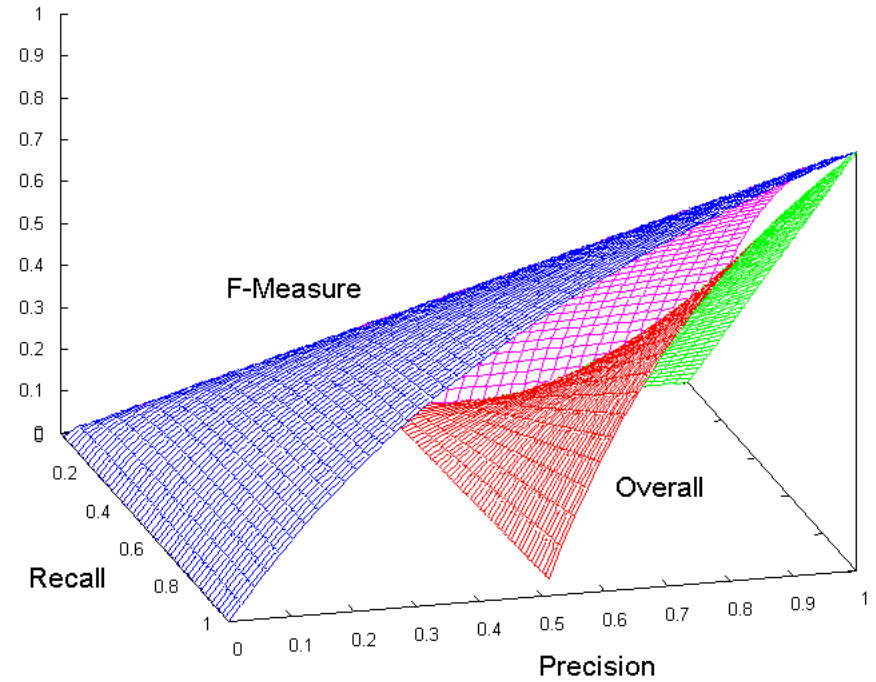
$$Precision = \frac{|B|}{|B| + |C|} \qquad Recall = \frac{|B|}{|A| + |B|}$$

Criteria: Quality Measures (cont.)

- Combined quality measures:
  - *F-Measure*: Harmonic mean (Information retrieval)
  - *Overall*: Manual effort to add missed and to remove false correspondences (Match context)

$$F\text{–}Measure = \frac{2 * |B|}{(|A| + |B|) + (|B| + |C|)} = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Overall = 1 - \frac{|A| + |C|}{|A| + |B|} = \frac{|B| - |C|}{|A| + |B|} = Recall * \left(2 - \frac{1}{Precision}\right)$$

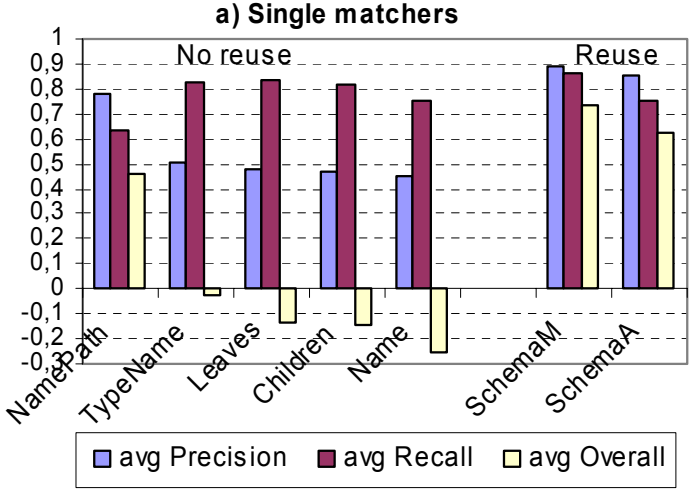


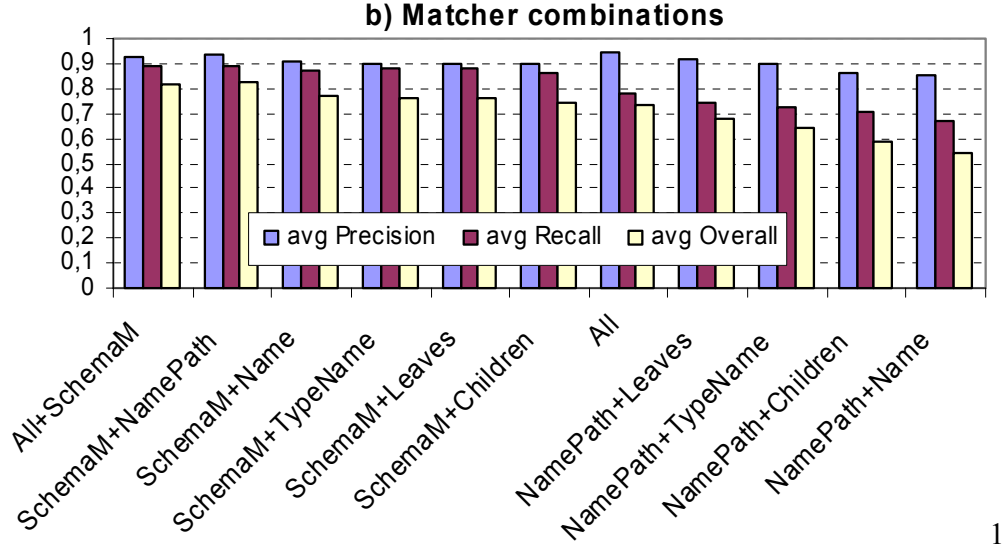
Criteria: Methodology

- Pre-match effort
  - Training machine learning-based matchers
  - Configuring match algorithms, e.g. setting weights, thresholds
  - Specifying auxiliary information, e.g. domain synonyms, constraints
- Post-match effort
  - Adding missed correspondences
  - Removing false correspondences
- Varying cost of manual operations
  - subjectivity, background knowledge of user, ...
- Largely unsolved problem !!!
  - Only post-match effort considered by combined measures

COMA Evaluation

- Composite approach
  - Comprehensive matcher library and various combination strategies
- Input
  - 5 XML schemas for purchase orders (biztalk.org), 10 match tasks
  - Domain synonyms
- Systematic evaluation of matchers and combination strategies
- Quality measures: *Precision, Recall, Overall*





Cupid Evaluation

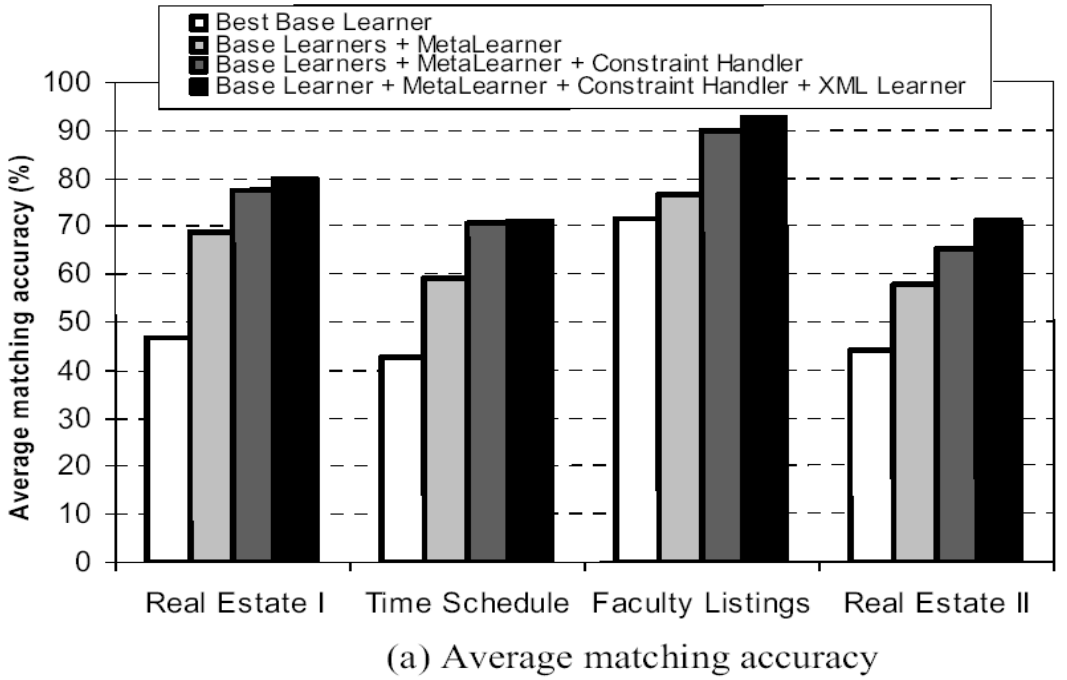
- Hybrid approach considering element names and schema structure
- Input:
  - Several small canonical examples + 2 real-world XML schemas
  - Domain synonyms
- Comparative evaluation of 3 match systems: Cupid, Dike, Momis
- No quality measures

CIDX → Excel	Cupid	DIKE	MOMIS – ARTEMIS
POHeader → Header	Yes	Yes	Yes
Item → Item	Yes	Yes	The two *Item* elements and the *Items* element are in a single cluster. *POLines* is in its own cluster.
POLines → Items	Yes	Yes	
POBillTo→InvoiceTo	Yes	No	Clustered together with the Address element
POShipTo→DeliverTo	Yes	No	
Contact→Contact	Yes	Yes	Yes
PO→PurchaseOrder	Yes	Yes	Yes, classes clustered, but corresponding elements not mapped.

Table 2: Mapping comparison for CIDX-EXCEL example

LSD Evaluation

- Composite approach combining different instance-level machine-learning matchers
- Input:
  - 4 domains: 20 sources + 4 global schemas, 20 match tasks
  - Domain synonyms and constraints
- Evaluation of different learner combinations
- Quality measures: *Recall*



(a) Average matching accuracy

Comparison Summary

		Autoplex/match	COMA	Cupid	LSD/GLUE	SemInt	SF
Input	Schemas/Tasks	15 Relational - 21 tasks	5 XML - 10 tasks	2 XML - 1 task	24 XML - 20 tasks	10 Relational - 5 tasks	18 XML+Rel. - 9 tasks
	Max size	15 / 11	145	54	66 / 333	260	22
Output	Matches	element-level correspondences with similarity values in range [0, 1]					
	Cardinality	1:1 leaf (attribute)	1:1 path		1:1 inner + leaf	attribute cluster (m:n)	1:1 inner + leaf
Methodology	Quality metrics	Precision, Recall / F-Measure	Precision, Recall, Overall	no	Recall	Precision, Recall	Overall
	Pre-match effort	training	domain synonyms		training, domain synonyms and constraints	none	none
	Impact on quality		matchers, combination strategies	comparison of 3 match systems	learner combinations	discriminators' sensitivity	user subjectivity
Best Average Quality	Precis.	considering all match results at once	0.92	not computed	only Recall used	clustering quality	-
	Recall		0.89				-
	F-Meas.		0.90				-
	Overall		0.82				~0.6

Conclusions

- Current situation
  - Various prototypes implementing a broad range of automatic schema matching techniques
  - Large differences in current evaluations
- Requirements for future systems and evaluations
  - Better conception and documentation for reproducibility
  - Input: Big schemas
  - Output: *Top-k* match suggestion
  - Quality measures: combined measures for more realistic quality assessment
  - Methodology: Systematic evaluation of system configurations, parameter settings
- Schema matching benchmark
  - Domain-specific
  - Match capabilities: name, structure, instance matching, etc.