A Scalable Approach for Large-Scale Schema Mediation

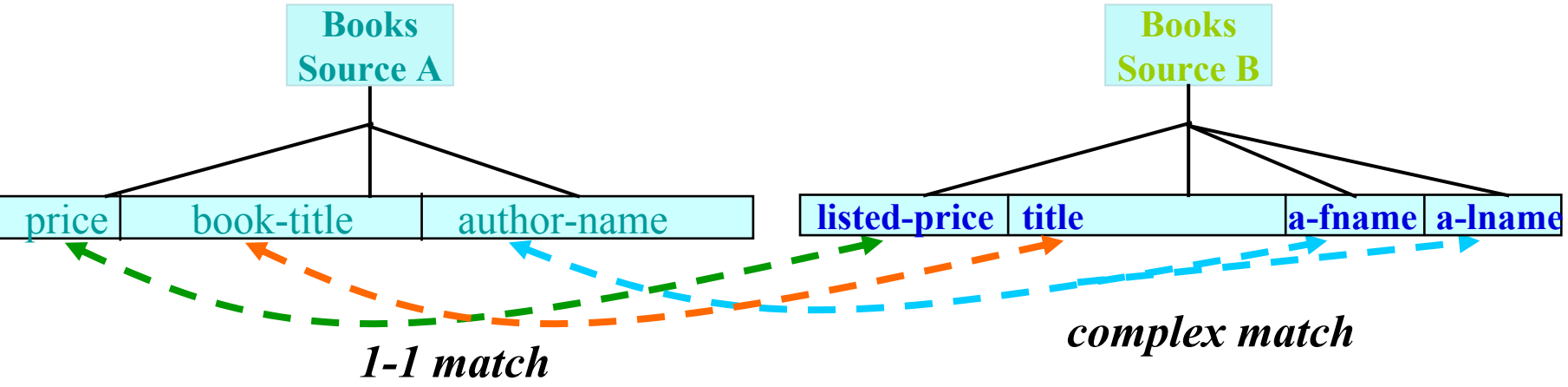
Khalid Saleem, Zohra Bellahsène
LIRMM CNRS/Université Montpellier 2, France

Outline

- Introduction
  - The matching problem
  - Brief state of the art
- A hybrid approach for large scale
  - Extracted from Tree Mining
  - Holistically exploits set of XML Schema trees
    - Each schema tree can have thousand of nodes
  - Promising, but still requires more work
- Approach applicable to other data models where metadata can have tree structure

Schema Matching

- Takes two schemas/ontologies as input and produces a mapping between elements of the two schemas that correspond *semantically* to each other



Brief State of the Art

- Schema Matching
  - Schema based : COMA, S-Match, Cupid …
  - Instance based : LSD, DUMAS …
- Ontology Matching : QOM, OLA, PROMPT …
- Use of External Oracles : Wordnet, SUMO, DOLCE, Domain specific global ontology
- Matching Systems Tuners : eTuner, OMEN
  - Match Results Tuners : [Manakanatas06], [Guadria07]

Quality vs Performance

- Semantic Match Quality is always approximate, normalized (0 – 1)

- Performance secondary objective
- Requires
  - Automated
  - domain specific, hybrid approach with
  - target search space optimization algorithms

Large-scale Schema Matching Problem

- Input
  - Large set of schemas (> 2)
  - Size of input schema is large (elements in 100s…)
- Output
  - Schema matching
  - Selecting the best match
  - Integrating the schemas
  - Schema mediation between input schemas and integrated schema (Mediated Schema)

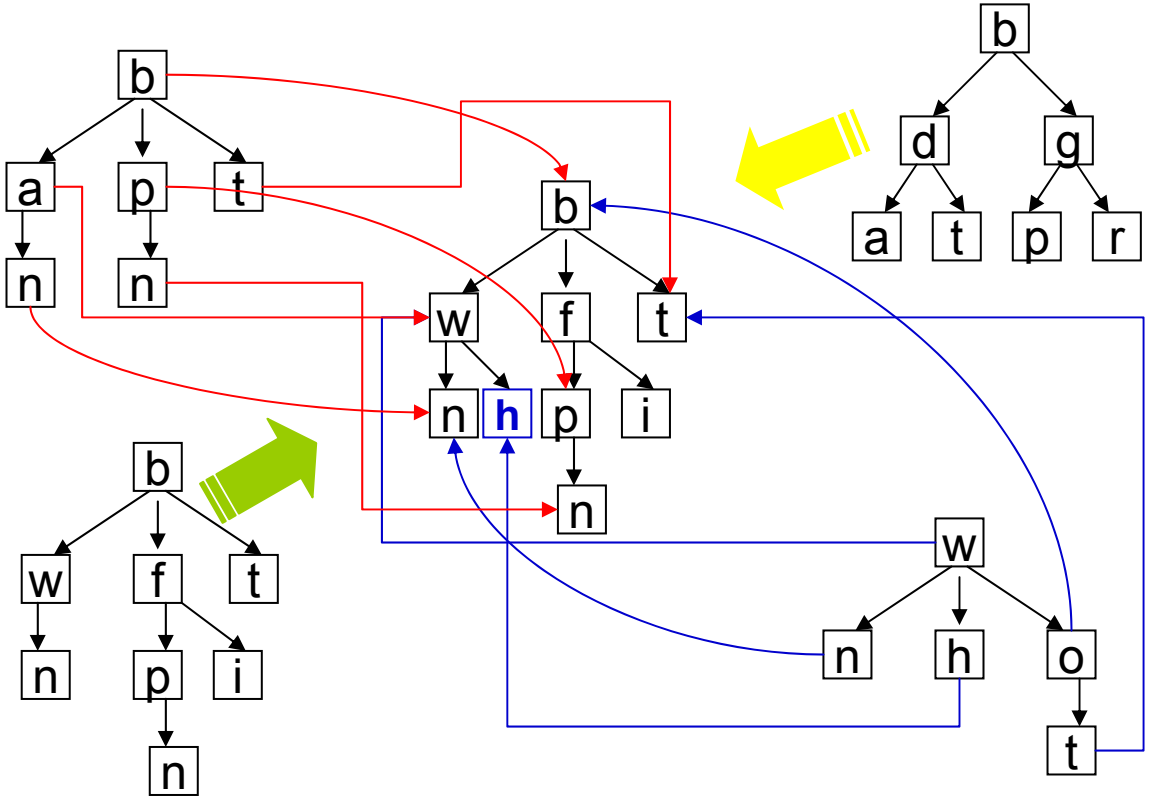
Large-scale Schema Matching

- Related Work
  - Large Taxonomy Matching
    - [Mork04], [Rahm05]
  - Holistic Matching
    - DCM [He04], PSM [Su06], [Wang04] …
  - Clustering
    - XClust[Lee02],[Smiljanic06] …

Our Approach …

- Assumptions
  - Schema are considered as trees
  - Input schema are domain specific
  - Semantically similar elements are rarely present in the same schema
    - Similar : author/ name = writer/ name
    - i.e. both represent same concept
    - Not Similar : author/ name = publisher/ name
  - Input schema tree with highest number of elements selected as initial mediated schema

An Example: Integrating more than 2 …



a: author
b: book
d: detail
f: information
g: general
h: birth
i: isbn
n: name
o: own-books
p: publisher
r: price
t: title
w: writer

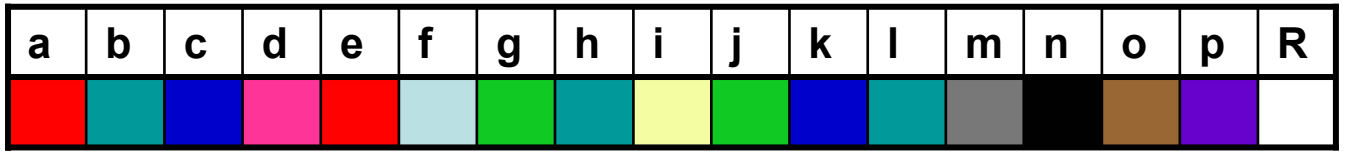
a=w
b=o
f=d

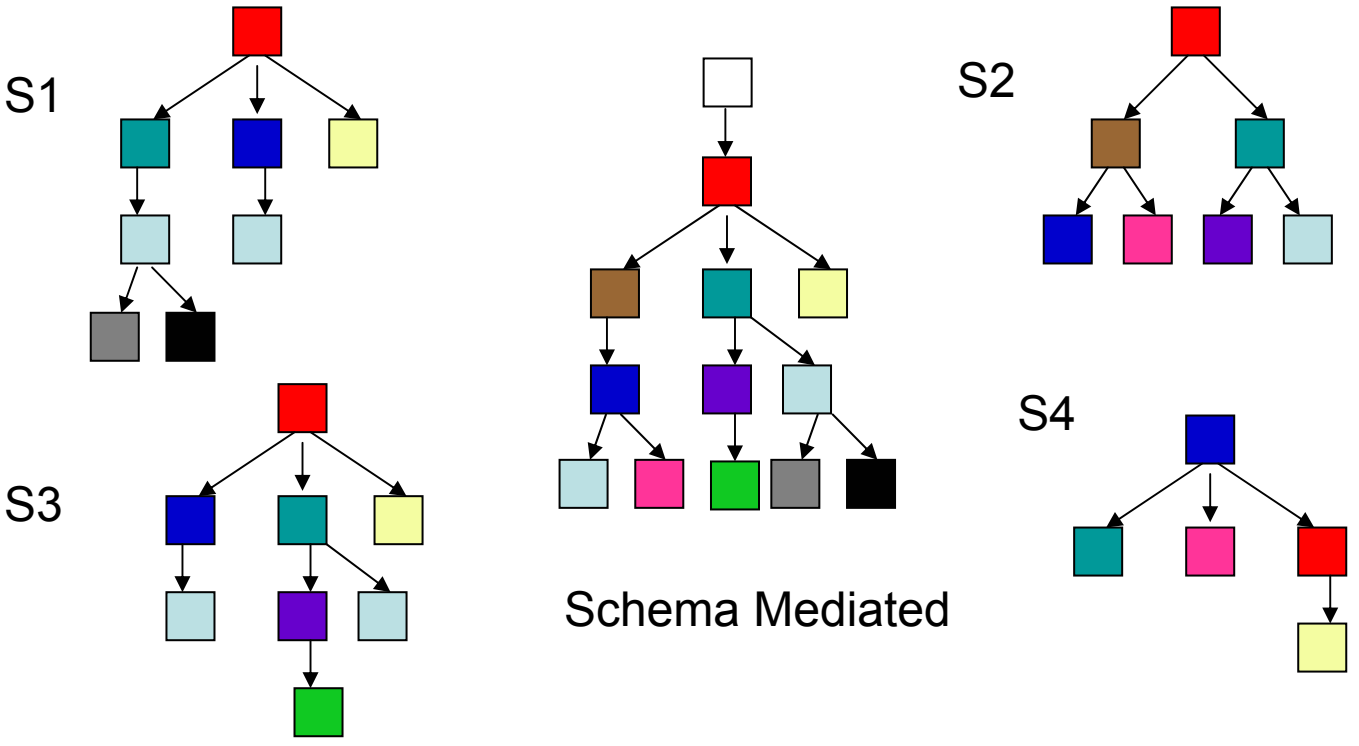
Our Approach : Key Idea

- Holistic
  - Analyse the whole set of schema trees
- Tree Mining [Zaki05]
  - Create distinct labels list in the set
  - Calculate pre-order for each node in respective tree
  - Calculate scope of each node
- Clustering
  - Cluster similar labels in the list of labels
  - Intuitively cluster possible similar nodes
- Context Similarity
  - 1:1 - Leaf to Leaf, Non-Leaf to Non-Leaf
  - 1:n – Leaf to Non-Leaf, n:1 – Non-Leaf to Leaf

Clustering

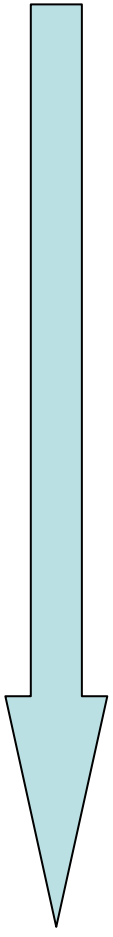
Label List - Same color for similar labels cluster





Implementation

- Node Analysis
  - Node Scope Calculation
  - Distinct Labels List
- Labels Analysis
  - Labels Abbreviation adjustment (Abbr. Table)
  - Labels Tokenization
  - Token Similarity (Synonym Table)
  - Similar Labels Clustering
- Node Mapping
  - Initial Mediated Schema Selection
  - Node Mapping
    - Target search space : Similar label nodes cluster in mediated schema
    - Node Context similarity verification using Scope Properties
      i.e. source and target nodes' ancestor/ parent nodes mapping exist or not …



Nodes Context Mining

- Using Scope Context Properties

- **Unary Properties,** given a node x [X,Y]

  Property 1. Leaf Node(x) : X=Y.

  Property 2. Non-Leaf Node(x): X<Y.

- **Binary Properties**

  Given x [X,Y], xd [Xd,Yd], xa [Xa,Ya], and xr[Xr,Yr].

  Property 3. Descendant (x,xd), xd is a descendant of x : Xd>X and Yd<=Y.

  Property 4. Descendant Leaf (x,xd) (combination of Property 1 and 3) : Xd>X and Yd ≤Y and Xd=Yd.

  Property 5. Ancestor (xa,a) (complement of Property 3) xa is ansector of x : Xa<X and Ya>=Y.

  Property 6. Right Hand Side Nodes with Non-Overlapping Scope : xr is Right Hand Side Node of x : Xr>Y.

Following by Example

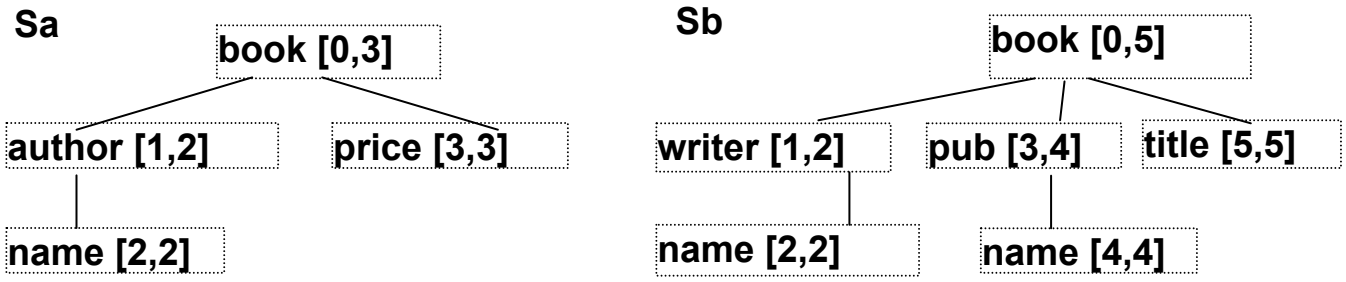


Table 1. Before NodeMapper Execution								
a. Labels List								
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
author	***book***	***name***	***name***	***price***	***pub***	***title***	***writer***	***ROOT***
b.Input Schema Nodes' Matrix								
1,2,0	***0,3,-1***	***2,2,1***		***3,3,0***				
	0,5,-1	***2,2,1***	***4,4,3***		***3,4,0***	***5,5,0***	***1,2,0***	
c. Initial Mediated Schema								
	1,6,0	***3,3,2***	***5,5,4***		***4,5,1***	***6,6,1***	***2,3,1***	***0,6,-1***

Example …

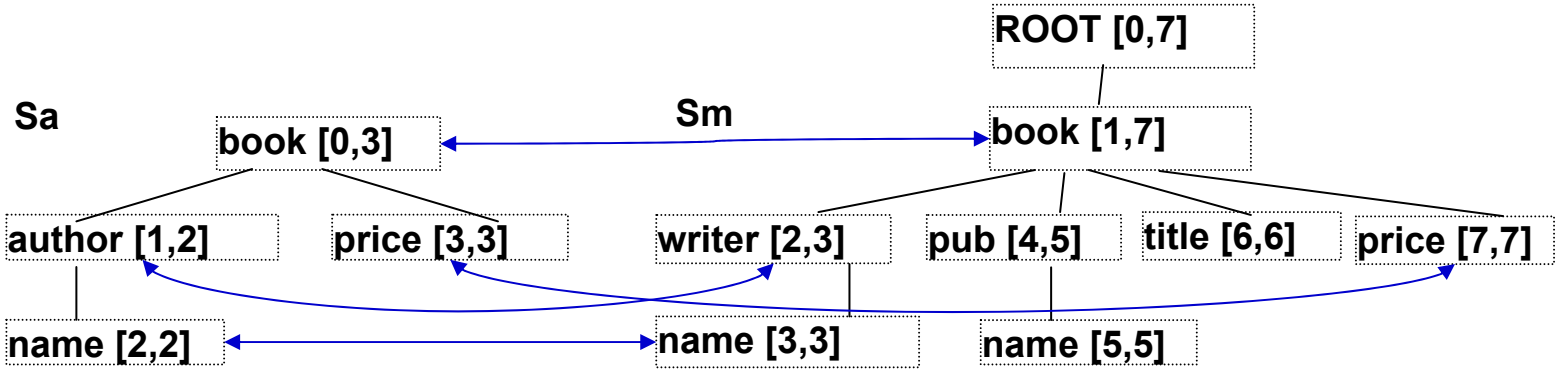


Table 3 . After NodeMapper Execution

a. Label List

0	1	2	3	4	5	6	7	8
author	*book*	*name*	*name*	*price*	*pub*	*title*	*writer*	*ROOT*

b. Mapping matrix

1,2,0 <7>	*0,3,-1 <1>*	*2,2,1 <2>*		*3,3,0 <4>*				
	0,5,-1 <1>	*2,2,1 <2>*	*4,4,3 <3>*		*3,4,0 <5>*	*5,5,0<6>*	*1,2,0<7>*	

c. Final Mediated Schema

	1,7,0,	*3,3,2*	*5,5,4*	*7,7,1*	*4,5,1*	*6,6,1*	*2,3,1*	*0,7,-1*
	1.0, 2.0	*1.2, 2.2*	*2.4*	*1.3*	*2.3*	*2.5*	*1.1, 2.1*	

Evaluation : Data Characteristics

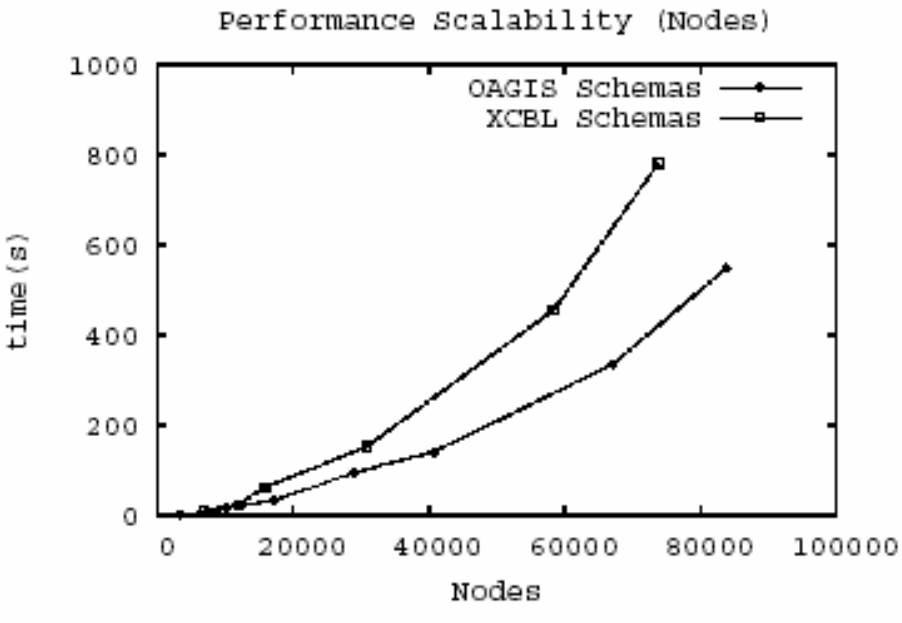
XML Schemas

	Domain 1 (Real) OAGIS	Domain 2 (Real) XCBL	Domain 3 (Synthetic) Books
Number of Schemas	80	44	176
Avg. nodes per schema	1047	1678	8
Largest/ smallest schema size	3519/ 26	4578/ 4	14/ 5

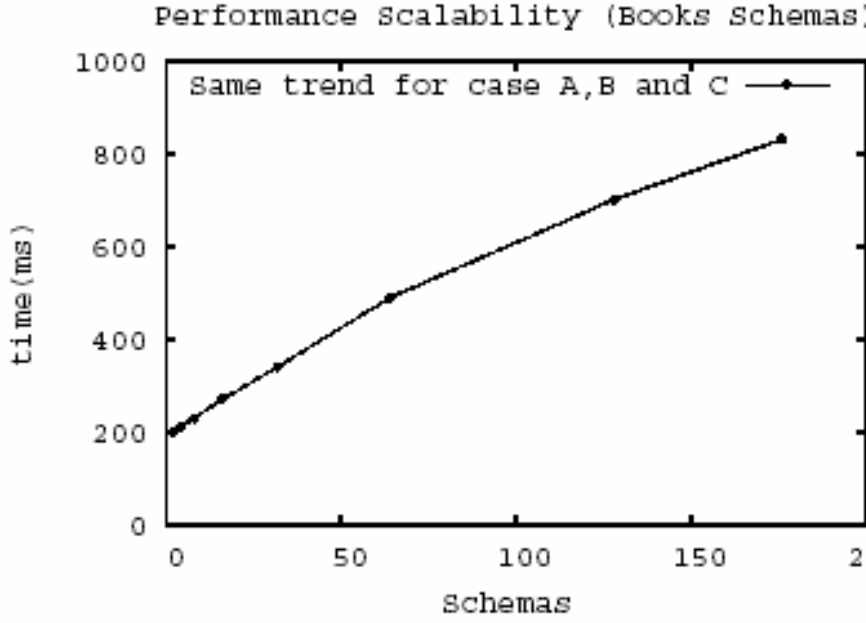
OAGIS : http://www.openapplications.org/
XCBL : http://www.xcbl.org/

Evaluation: Performance

A) Label String Equivalence
B) Token Set Equivalence
C) Synonym Token Set Equivalence



Comparison of schema integration times for real web schemas

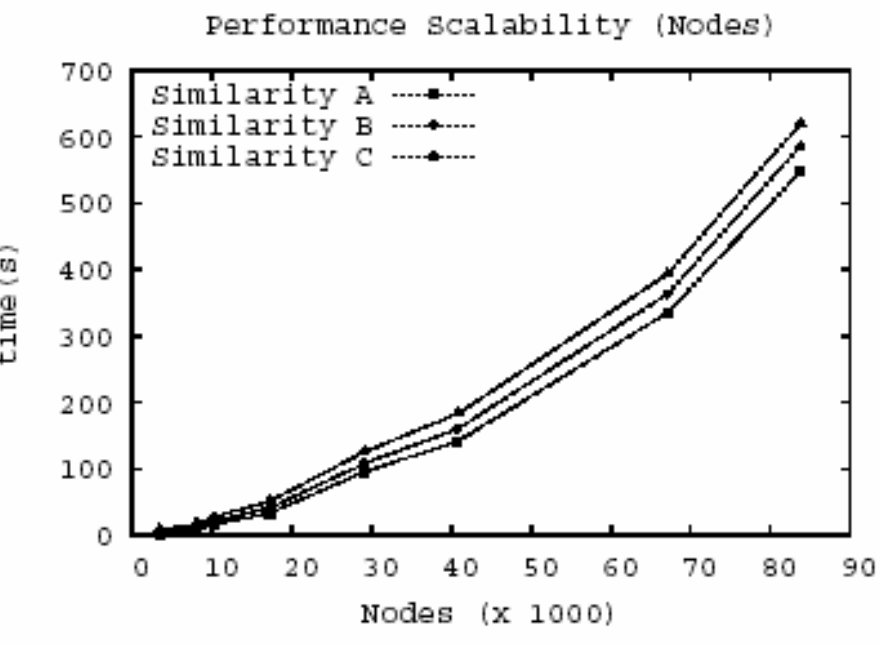


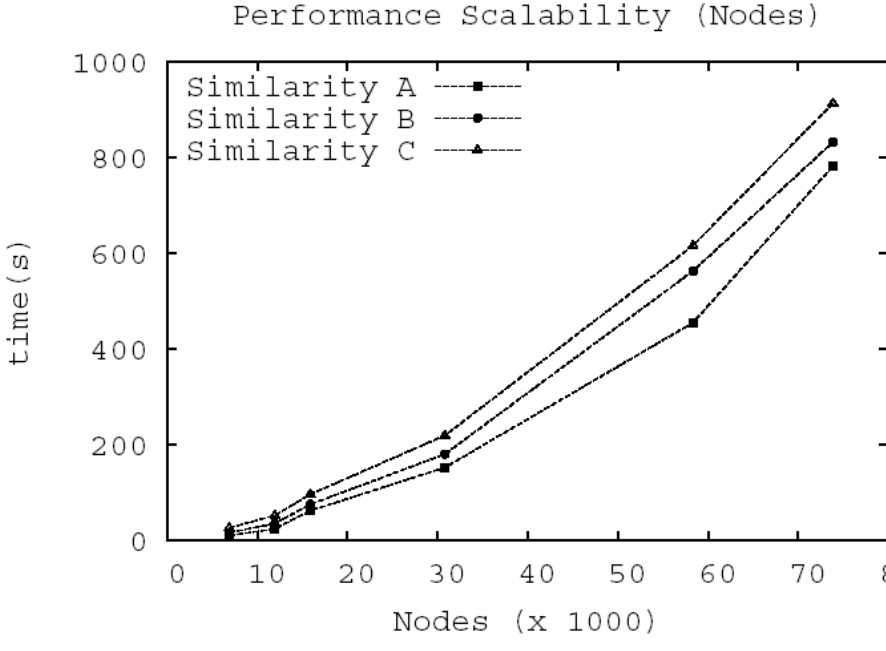
Integration time with reference to the number of schemas in BOOKS

The time is directly proportional to the number of nodes processed

Evaluation: Performance

A) Label String Equivalence
B) Token Set Equivalence
C) Synonym Token Set Equivalence





The time is directly proportional to the number of distinct labels (tokens list)

Evaluation: Match Quality

Domain / Schema Size / Match Tool	Purchase Order		Books		OAGIS	
	S1 18	S2 14	S1 15	S2 12	S1 2931	S2 475
	Time	Qlty	Time	Qlty	Time	Qlty
Our approach	0.2	=	0.2	=	2.5	---
COMA++	5	=	3	=	370	---

The abbreviation and synonym tables used were related to Purchase order and Books domain

Concluding Remarks and Future Work

- Performance is crucial in large scale schema matching and integration
- Provides a hybrid automatic solution
  - Flexible enough to add more label similarity measures at the cost of performance
  - Simple scope related integer computation for context mining
  - Structural context match is semantic match
- Extensive experiments over 2 real domains and 1 synthetic domain
- Future directions
  - Optimize implementation data structure
  - Apply to other data models (converted to trees)
  - Enhance this technique to calculate n:m matches and implement n:m mappings with the mediated schema

Some References

Ameueller05] D. Aumueller, H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with coma++. In *SIGMOD* 2005

Do05] H. H. Do. Schema Matching and Mapping-Based Data Integration. PhD thesis, University of Leipzig, 2005.

Euzenat04] J. Euzenat et al. State of the art on ontology matching. Technical Report KWEB/2004/D2.2.3/v1.2, *Knowledge Web*, 2004

Giunchiglia05] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *Semantic Interoperability and Integration*, 2005

Guadria07] W. Guedria, Z. Bellahs´ene, and M. Roche. A flexible approach based on the user preferences for schema matching. In *RCIS*, 2007

He04] B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across webquery interfaces: a correlation mining approach. In *KDD*, pages 148–157, 2004.

Lee02] M.-L. Lee, L. H. Yang, W. Hsu, and X. Yang. Xclust: clustering xml schemas for effective integration. In *CIKM*, pages 292–299, 2002

Lee07] Y. Lee, M. Sayyadain, A. Doan, and A. S. Rosenthal. etuner: tuning schema matching software using synthetic scenarios. *VLDB Journal*, 16:97–122, 2007

Madhavan01] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, 2001

Manakanatas06] D. Manakanatas, D. Plexousakis .A Tool for Semi-Automated Semantic Schema Mapping: Design and Implementation. In *DisWEB Workshop, CaiSE* 2006.

Mitra05] P. Mitra, N. F. Noy, and A. R. Jaiswal. Omen: A probabilistic ontology mapping tool. In *International Semantic Web Conference*, pages 537–547, 2005

Mork04] P. Mork and P. A. Bernstein. Adapting a generic match algorithm to align ontologies of human anatomy. In *ICDE*, 2004

Rahm04] E. Rahm, H. H. Do, and S. Massmann. Matching large xml schemas. *SIGMOD Record*, 33(4):26–31, 2004

Smiljanic06] M. Smiljanic, M. van Keulen, and W. Jonker. Using element clustering to increase the efficiency of xml schema matching. In *Workshop ICDE*, 2006

Su06] W. Su, J.Wang, and F. Lochovsky. Holistic query interface matching using parallel schema matching. In *ICDE*, 2006.

Wang04] J. Wang, J. Wen, F. Lochovsky, and W. Ma. Instance-based schema matching for web databases by domain-specific query probing. In *VLDB*, 2004.

Zaki05] Zaki, M.J. Efficiently Mining Frequent Embedded Unordered Trees. *Fundamenta Informaticea*, 2005