

Schema Matching

Craig Knoblock

Based on slides by AnHai Doan

Learning Models of the Sources: Source Modeling vs. Schema Matching

- Schema Matching
 - Align schemas between data sources
 - Assumes static sources and complete access to data
- Source modeling
 - Incrementally build models from partial data (e.g., web services, html forms, programs)
 - Model not just the fields but the source types and even the function of a source
 - Support richer source models (a la Semantic Web)

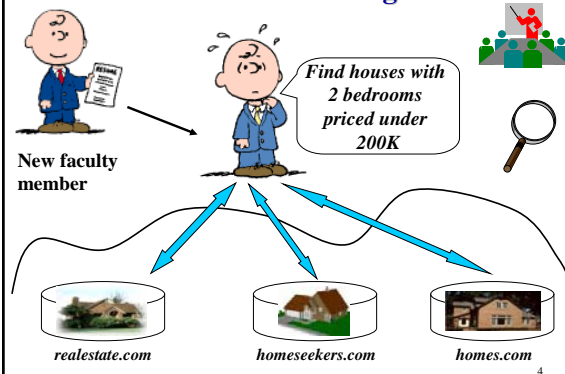
2

Road Map

- ➔ • Schema matching motivation & problem definition
- Representative current solutions: LSD, iMAP, Clio
- Broader picture and conclusions

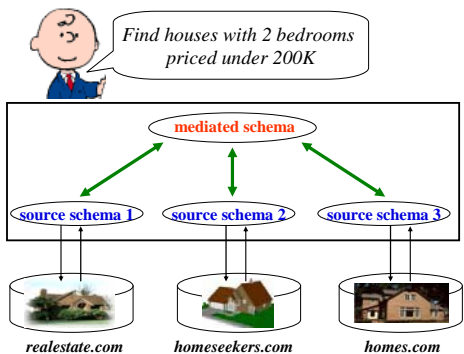
3

Motivation: Data Integration



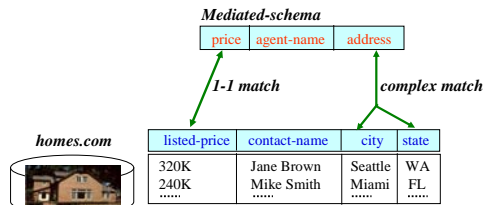
4

Architecture of Data Integration System



5

Semantic Matches between Schemas



6

Schema Matching is Ubiquitous!

- **Fundamental problem in numerous applications**
- Databases
 - data integration
 - data translation
 - schema/view integration
 - data warehousing
 - semantic query processing
 - model management
 - peer data management
- AI
 - knowledge bases, ontology merging, information gathering agents, ...
- Web
 - e-commerce
 - marking up data using ontologies (e.g., on Semantic Web)

7

Why Schema Matching is Difficult

- Schema & data never fully capture semantics!
 - not adequately documented
 - schema creator has retired to Florida!
- Must rely on clues in schema & data
 - using names, structures, types, data values, etc.
- Such clues can be unreliable
 - same names => different entities: **area** => **location** or **square-feet**
 - different names => same entity: **area** & **address** => **location**
- Intended semantics can be subjective
 - **house-style** = **house-description**?
 - military applications require committees to decide!
- Cannot be *fully* automated, needs user feedback!

8

Current State of Affairs

- **Finding semantic mappings is now a key bottleneck!**
 - largely done by hand
 - labor intensive & error prone
 - data integration at GTE [Li&Clifton, 2000]
 - 40 databases, 27000 elements, estimated time: 12 years
- Will only be exacerbated
 - data sharing becomes pervasive
 - translation of legacy data
- **Need semi-automatic approaches to scale up!**
- Many research projects in the past few years
 - **Databases**: IBM Almaden, Microsoft Research, BYU, George Mason, U of Leipzig, U Wisconsin, NCSU, UIUC, Washington, ...
 - **AI**: Stanford, Karlsruhe University, NEC Japan, ...

9

Road Map

- Schema matching motivation & problem definition
- ➔ ● Representative current solutions: **LSD**, **iMAP**, **Clio**
- Broader picture and conclusions

10

LSD

- Learning Source Description
- Developed at Univ of Washington 2000-2001
 - with Pedro Domingos and Alon Halevy
- Designed for data integration settings
 - has been adapted to several other contexts
- Desirable characteristics
 - learn from **previous matching activities**
 - exploit **multiple types of information in schema and data**
 - incorporate **domain integrity constraints**
 - handle **user feedback**
 - achieves high matching accuracy (66 -- 97%) on real-world data

11

Schema Matching for Data Integration: the LSD Approach

Suppose user wants to integrate 100 data sources

1. User
 - **manually creates** matches for **a few** sources, say 3
 - shows **LSD** these matches
2. **LSD learns** from the matches
3. **LSD predicts** matches for remaining 97 sources

12

Learning from the Manual Matches

Mediated schema

If "office" occurs in name => office-phone

Schema of realestate.com

listed-price	contact-name	contact-phone	office	comments
\$250K	James Smith	(305) 729 0831	(305) 616 1822	Fantastic house
\$320K	Mike Doan	(617) 253 1429	(617) 112 2315	Great location
.....

Schema of homes.com

sold-at	contact-agent	extra-info
\$350K	(206) 634 9435	Beautiful yard
\$230K	(617) 335 4243	Close to Seattle

If "fantastic" & "great" occur frequently in data instances => description

13

Must Exploit Multiple Types of Information!

Mediated schema

If "fantastic" & "great" occur frequently in data instances => description

Schema of realestate.com

listed-price	contact-name	contact-phone	office	comments
\$250K	James Smith	(305) 729 0831	(305) 616 1822	Fantastic house
\$320K	Mike Doan	(617) 253 1429	(617) 112 2315	Great location
.....

Schema of homes.com

sold-at	contact-agent	extra-info
\$350K	(206) 634 9435	Beautiful yard
\$230K	(617) 335 4243	Close to Seattle

If "fantastic" & "great" occur frequently in data instances => description

14

Multi-Strategy Learning

- Use a set of **base learners**
 - each exploits well certain types of information
- To match a schema element of a new source
 - apply base learners
 - combine their predictions using a **meta-learner**
- Meta-learner
 - uses **training sources** to measure base learner accuracy
 - weighs each learner based on its accuracy

15

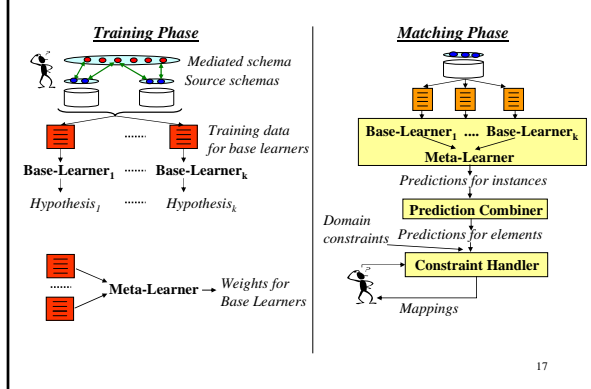
Base Learners

- Training**
- Matching**

$X \Rightarrow$ labels weighted by confidence score
- Name Learner**
 - training: ("location", address), ("contact name", name),
 - matching: agent-name \Rightarrow (name,0.7),(phone,0.3)
- Naive Bayes Learner**
 - training: ("Seattle, WA", address), ("250K", price),
 - matching: "Kent, WA" \Rightarrow (address,0.8),(name,0.2)

16

The LSD Architecture



Training the Base Learners

Mediated schema

If "fantastic" & "great" occur frequently in data instances => description

realestate.com

location	price	contact-name	contact-phone	office	comments
Miami, FL	\$250K	James Smith	(305) 729 0831	(305) 616 1822	Fantastic house
Boston, MA	\$320K	Mike Doan	(617) 253 1429	(617) 112 2315	Great location
.....

homes.com

sold-at	contact-agent	extra-info
\$350K	(206) 634 9435	Beautiful yard
\$230K	(617) 335 4243	Close to Seattle

Name Learner

- ("location", address)
- ("price", price)
- ("contact name", agent-name)
- ("contact phone", agent-phone)
- ("office", office-phone)
- ("comments", description)

Naive Bayes Learner

- ("Miami, FL", address)
- ("\$250K", price)
- ("James Smith", agent-name)
- ("(305) 729 0831", agent-phone)
- ("(305) 616 1822", office-phone)
- ("Fantastic house", description)
- ("Boston,MA", address)
-

18

Meta-Learner: Stacking

[Wolpert 92, Ting & Witten 99]

- Training
 - uses training data to learn weights
 - one for each (base-learner, mediated-schema element) pair
 - weight (Name-Learner, address) = 0.2
 - weight (Naive-Bayes, address) = 0.8
- Matching: combine predictions of base learners
 - computes **weighted average** of base-learner confidence scores

area	Name Learner	→ (address, 0.4)
Seattle, WA	Naive Bayes	→ (address, 0.9)
Kent, WA		
Bend, OR		
Meta-Learner		→ (address, 0.4*0.2 + 0.9*0.8 = 0.8)

19

The LSD Architecture

Training Phase

Matching Phase

20

Applying the Learners

homes.com schema

area	sold-at	contact-agent	extra-info				
<table border="1" style="width: 100%;"> <tr><td>area</td></tr> <tr><td>Seattle, WA</td></tr> <tr><td>Kent, WA</td></tr> <tr><td>Bend, OR</td></tr> </table>	area	Seattle, WA	Kent, WA	Bend, OR
area							
Seattle, WA							
Kent, WA							
Bend, OR							
<table border="1" style="width: 100%;"> <tr><td>area</td></tr> <tr><td>Seattle, WA</td></tr> <tr><td>Kent, WA</td></tr> <tr><td>Bend, OR</td></tr> </table>	area	Seattle, WA	Kent, WA	Bend, OR
area							
Seattle, WA							
Kent, WA							
Bend, OR							
.....				
.....				
.....				

.....
.....
.....
.....
.....

21

Domain Constraints

- Encode user knowledge about domain
- Specified **only once**, by examining mediated schema
- Examples
 - at most one source-schema element can match **address**
 - if a source-schema element matches **house-id** then it is a key
 - avg-value(price) > avg-value(num-baths)
- Given a mapping combination
 - can verify if it satisfies a given constraint

area:	address
sold-at:	price
contact-agent:	agent-phone
extra-info:	address

22

The Constraint Handler

Predictions from Prediction Combiner

area:	(address, 0.7), (description, 0.3)
sold-at:	(price, 0.9), (agent-phone, 0.1)
contact-agent:	(agent-phone, 0.9), (description, 0.1)
extra-info:	(address, 0.6), (description, 0.4)

Domain Constraints

At most one element matches **address**

area:	address	0.7	0.3
sold-at:	price	0.9	0.1
contact-agent:	agent-phone	0.9	0.1
extra-info:	address	0.6	0.4
		0.3402	0.2268	0.0012

- Searches space of mapping combinations efficiently
- Can handle arbitrary constraints
- Also used to incorporate user feedback
 - **sold-at** does not match **price**

23

The Current LSD System

- Can also handle data in XML format
 - matches XML DTDs
- Base learners
 - **Naive Bayes** [Duda & Hart-93, Domingos & Pazzani-97]
 - exploits frequencies of words & symbols
 - **WHIRL Nearest-Neighbor Classifier** [Cohen & Hirsh KDD-98]
 - employs information-retrieval similarity metric
 - **Name Learner** [SIGMOD-01]
 - matches elements based on their names
 - **County-Name Recognizer** [SIGMOD-01]
 - stores all U.S. county names
 - **XML Learner** [SIGMOD-01]
 - exploits hierarchical structure of XML data

24

Empirical Evaluation

- Four domains
 - Real Estate I & II, Course Offerings, Faculty Listings
- For each domain
 - created mediated schema & domain constraints
 - chose five sources
 - extracted & converted data into XML
 - mediated schemas: 14 - 66 elements, source schemas: 13 - 48
- Ten runs for each domain, in each run:
 - manually provided 1-1 matches for 3 sources
 - asked LSD to propose matches for remaining 2 sources
 - accuracy = % of 1-1 matches correctly identified

25

High Matching Accuracy

LSD's accuracy: 71 - 92%

□ Best single base learner:	42 - 72%
■ + Meta-learner:	+ 5 - 22%
■ + Constraint handler:	+ 7 - 13%
■ + XML learner:	+ 0.8 - 6%

26

Contribution of Schema vs. Data

More experiments in [Doan et al. SIGMOD-01]

27

LSD Summary

- LSD
 - learns from previous matching activities
 - exploits multiple types of information
 - by employing multi-strategy learning
 - incorporates domain constraints & user feedback
 - achieves high matching accuracy
- LSD focuses on 1-1 matches
- Next challenge: discover more complex matches!
 - iMAP (Illinois Mapping) system [SIGMOD-04]
 - developed at Washington and Illinois, 2002-2004
 - with Robin Dhamanka, Yoonkyong Lee, Alon Halevy, Pedro Domingos

28

The iMAP Approach

listings	price	num-baths	address
homes.com	price	num-baths	address
listings	agent-id	full-baths	half-baths
listings	city	zipcode	
320K	53211	2	1
240K	11578	1	1
.....
			Seattle
			Miami
			23591
		

- For each mediated-schema element
 - searches space of all matches
 - finds a small set of likely match candidates
 - uses LSD to evaluate them
- To search efficiently
 - employs a specialized searcher for each element type
 - Text Searcher, Numeric Searcher, Category Searcher, ...

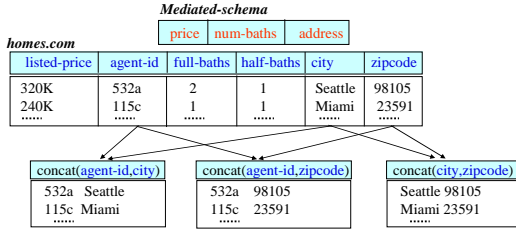
29

The iMAP Architecture [SIGMOD-04]

30

An Example: Text Searcher

- **Beam search** in space of all concatenation matches
- Example: find match candidates for **address**



- Best match candidates for **address**
 - (agent-id,0.7), (concat(agent-id,city),0.75), (concat(city,zipcode),0.9)

31

Empirical Evaluation

- Current **iMAP** system
 - 12 searchers
- Four real-world domains
 - real estate, product inventory, cricket, financial wizard
 - target schema: 19 -- 42 elements, source schema: 32 -- 44
- Accuracy: 43 -- 92%
- Sample discovered matches
 - **agent-name** = concat(first-name,last-name)
 - **area** = building-area / 43560
 - **discount-cost** = (unit-price * quantity) * (1 - discount)
- More detail in [Dhamanka et. al. SIGMOD-04]

32

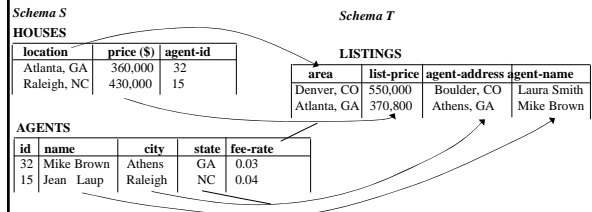
Observations

- Finding complex matches much harder than 1-1 matches!
 - require gluing together many components
 - e.g., num-rooms = bath-rooms + bed-rooms + dining-rooms + living-rooms
 - if missing one component => incorrect match
- However, even **partial matches** are already very useful!
 - so are top-k matches => need methods to **handle partial/top-k matches**
- Huge/infinite search spaces
 - **domain knowledge** plays a crucial role!
- Matches are fairly complex, hard to know if they are correct
 - must be able to **explain matches**
- Human must be fairly active in the loop
 - need **strong user interaction facilities**
- **Break matching architecture into multiple "atomic" boxes!**

33

Finding Matches is only Half of the Job!

- To translate data/queries, need **mappings**, not **matches**



- **Mappings**
 - **area** = SELECT location FROM HOUSES
 - **agent-address** = SELECT concat(city,state) FROM AGENTS
 - **list-price** = price * (1 + fee-rate)
FROM HOUSES, AGENTS
WHERE agent-id = id

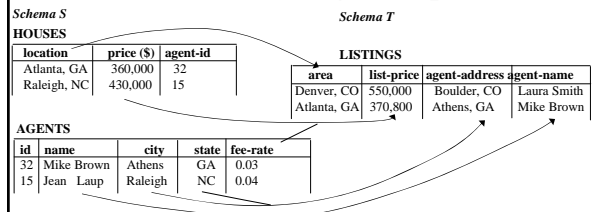
34

Clio: Elaborating Matches into Mappings

- Developed at Univ of Toronto & IBM Almaden, 2000-2003
 - by Renee Miller, Laura Haas, Mauricio Hernandez, Lucian Popa, Howard Ho, Ling Yan, Ron Fagin
- Given a match
 - list-price = price * (1 + fee-rate)
- Refine it into a mapping
 - list-price = SELECT price * (1 + fee-rate)
FROM HOUSES (FULL OUTER JOIN) AGENTS
WHERE agent-id = id
- Need to discover
 - the correct join path among tables, e.g., agent-id = id
 - the correct join, e.g., full outer join? inner join?
- Use heuristics to decide
 - when in doubt, ask users
 - employ sophisticated user interaction methods [VLDB-00, SIGMOD-01]

35

Clio: Illustrating Examples



- **Mappings**
 - **area** = SELECT location FROM HOUSES
 - **agent-address** = SELECT concat(city,state) FROM AGENTS
 - **list-price** = price * (1 + fee-rate)
FROM HOUSES, AGENTS
WHERE agent-id = id

36

Road Map

- Schema matching motivation & problem definition
- Representative current solutions: LSD, iMAP, Clio
- ➔ • Broader picture and conclusions

37

Broader Picture: Find Matches

*Hand-crafted rules
Exploit schema
1-1 matches*

TRANSCM [Milo&Zohar98]
ARTEMIS [Castano&Antonellis99]
[Palopoli et al. 98]
CUPID [Madhavan et al. 01]

*Single learner
Exploit data
1-1 matches*

SEMINT [Li&Clifton94]
ILA [Perkowitz&Etzioni95]
DELTA [Clifton et al. 97]
AutoMatch, Autoplex
[Berlin & Motro, 01-03]

Learners + rules, use multi-strategy learning

*Exploit schema + data
1-1 + complex matches*

Exploit domain constraints

LSD [Doan et al., SIGMOD-01]
iMAP [Dhamanka et al., SIGMOD-04]

Other Important Works

COMA by Erhard Rahm group
David Embley group at BYU
Jaewoo Kang group at NCSU
Kevin Chang group at UIUC
Clement Yu group at UIC

More about some of these works soon

38

Broader Picture: From Matches to Mappings

Rules

Exploit data

Powerful user interaction

CLIO [Miller et al., 00]
[Yan et al. 01]

Learners + rules

Exploit schema + data

1-1 + complex matches

Automate as much as possible

iMAP [Dhamanka et al., SIGMOD-04]

?

39

Need Much More Domain Knowledge

- Where to get it?
 - past matches (e.g., LSD, iMAP)
 - other schemas in the domain
 - holistic matching approach by Kevin Chang group [SIGMOD-02]
 - corpus-based matching by Alon Halevy group [IJCAI-03]
 - clustering to achieve bridging effects by Clement Yu group [SIGMOD-04]
 - external data (e.g., iMAP at SIGMOD-04)
 - mass of users (e.g., MOBS at WebDB-03)
- How to get it and how to use it?
 - no clear answer yet

40

Summary

- Schema/ontology matching:
 - key to numerous data management problems
 - much attention in the database, AI, Semantic Web communities
- Simple problem definition, yet very difficult to do
 - no satisfactory solution yet
 - AI complete?
- We now understand the problems much better
 - still at the beginning of the journey
 - will need techniques from multiple fields

41