Web-scale Data Integration: You can only afford to Pay As You Go

---- Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin (Luna) Dong, David Ko, Cong Yu, Alon Halevy, Google, Inc.

&

Bootstrapping Pay-As-You-Go Data Integration Systems

---- Anish Das Sarma, Xin Dong, Alon Halevy

Vishrawas Gopalakrishnan

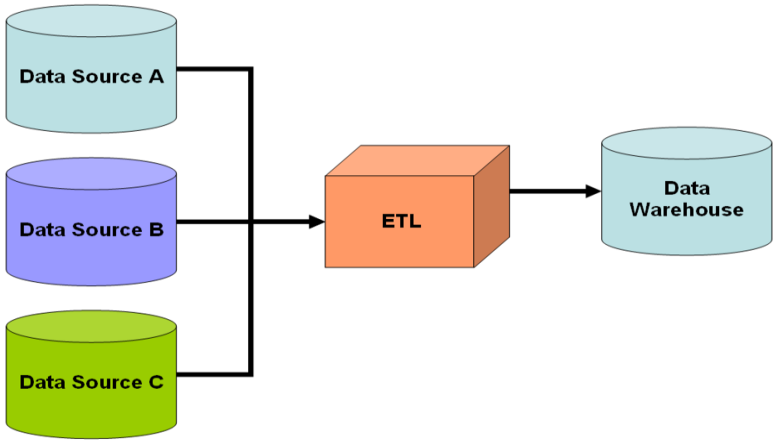
vishrawa@buffalo.edu

What is today's topic About?

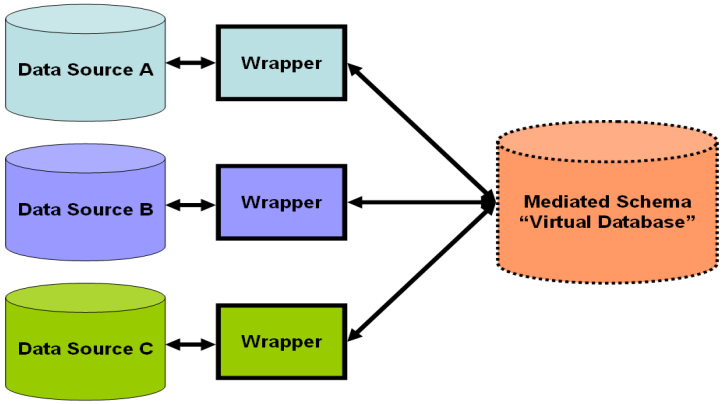
- Pay-As-You-Go-Data Integration System.
- Why Only Pay-As-You-Go In Web ?
- How To Bootstrap Pay-As-You-Go Data Integration System.

What is a Mediated Schema ?

- Mediated Schema – Nothing but a virtual schema



A traditional ETL Data warehouse scheme



An Equivalent Data Integration Scheme

For today the area of interest lies in Mediated schema

Structured Data on the Web

- World Wide Web is becoming structured
  - Deep Web
  - Google Base
  - Flickr
- How best can web-search handle structured data?
  - How can we search over structured data sources?
  - Can being structure-aware enhance web-search?
  - Or are we doomed to use traditional IR method?
- Heterogeneity of Data.

Paper 1: Approach

Discusses:

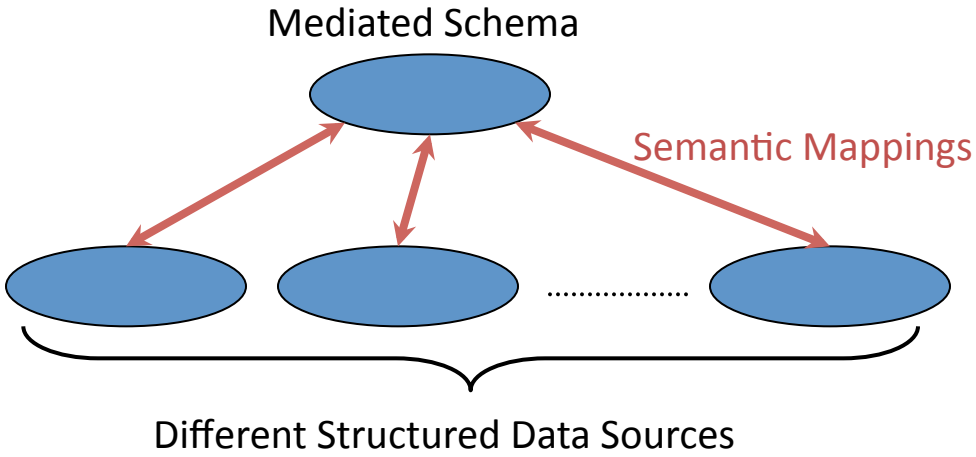
- Problems in approach towards Deep web:
  - *run-time query reformulation.*
  - *deep-web surfacing.*
- Google Base – show how schema is useful in enhancing user's search
- Briefly touch upon annotation schemes

Why Web-scale integration is PAYGO

- When it comes to web we need to model everything!
- We cannot model a domain or a set of domain because of the heterogeneity of the content
- Hence no well designed schema.
- Web Scale integration itself is pay-as-you-go

Typical Data Integration Solution

- Setting up integration systems
  - Design a mediated schema
  - Create semantic mappings



- Answering queries
  - Reformulate query over mediated schema into queries over data sources
  - Retrieve results from data sources and combine results

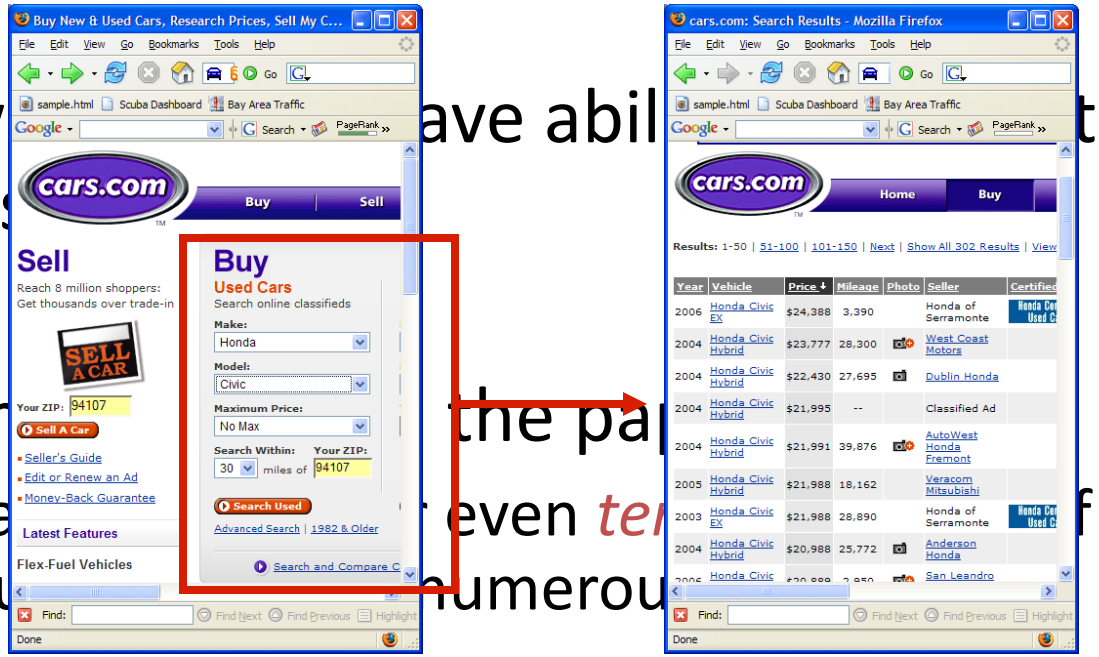
- Does not generalize well on a web-scale
  - Nature of structured data – quantity, heterogeneity, user queries

What Is PAYGO

- Creation of *on-the-fly* integration.
- System Starts with very few semantic mapping.
- Improve on these mappings as system progresses.

Deep Web

- Data that lies in backend databases that are only accessible through HTML forms
- Craw ave abil tary HTML form
- Exter the pa
  - Ma even *te* f data sou umerou

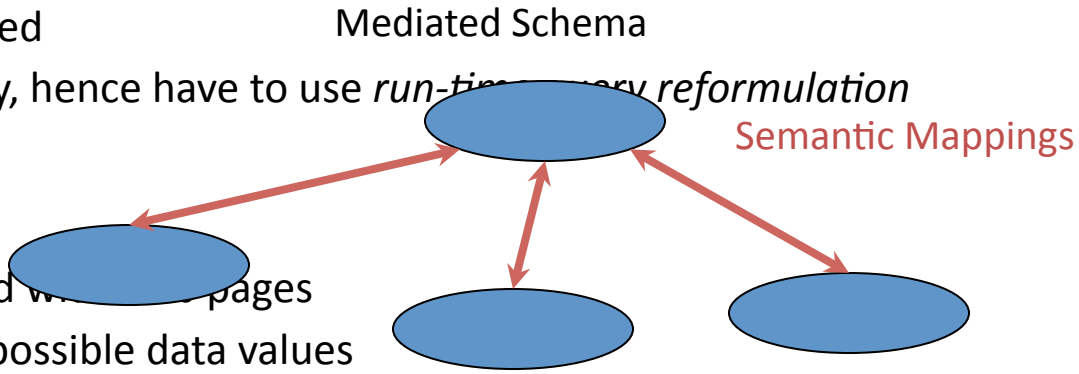


Indexing Deep Web

- Create Virtual Schema for a particular domain

  Problems
  - Large number of domains
  - Amount of information carried
  - Reliance on structured query, hence have to use *run-time query reformulation*
  - *Deep-web surfacing.*

  Problems:
  - Loss of semantics associated with [illegible] pages
  - Not easy to enumerate the possible data values



- Ideal Solution:

  Identify right sources that are likely to have relevant results, reformulate the query into a structured query over the relevant sources, retrieve the results and present them to the user i.e *query routing*

Google Base

- Semi-structured data uploaded to Google

HONDA CIVIC 2002

Price: $11,900.00 Model: civic Color: red Year: 2,002 Mileage: 56,247 Condition: used
HONDA CIVIC 2002 WORCESTER MA 01606 RED.
http://wwww.getauto.com - from GetAuto.com on Jan 8 - Report item

- Structure-awareness enhances search in Google Base
- a *very large, self-describing, semi-structured, heterogeneous database yet self describing*

- Demonstrates large scale heterogeneity
  - Large number of item types (more than 10,000)

Vehicles, Jobs, …, High Performance Car Parts, Marine Engine Parts

Google Base

Challenges faced in Google Base:

- Complexity of handling large number of item types.
- Issues related to schema management:
  - Specialization Hierarchy.
  - Heterogeneity caused by “User”.

Querying Google Base

Challenges faced:

- Query routing to determine relevant item types.

- Query refinement to interactively construct well-specified structured queries

Illustrations

1. user specifies a particular item type and perhaps provides values for some of the attributes(query refinements by computing histograms on attributes and their values during query time)
2. keyword query over *all of Google Base.*
3. keyword query on the main search engine, google.com

So what did We Learn?

- Structure helps.
- But you should have complete knowledge of the structure.
- So incase of web what we have to do ??

So what did We Learn?

- Incorpor
  descripti ntents.

Structured Data helps in querying but..

Difficulty?

Exasperate ges that are in evidence in Google Base.

So what did We Learn?

- Structured Data will be heterogeneous
- Web is about everything.
- No c

or ra le

and hard to maintain

Then Do What?

Moral :

- Current data integration architectures cannot cope with this web-scale heterogeneity.

PAYGO Architecture

- There can be many, potentially ill-defined, domains

  Mediated Schema → ***Schema Clusters***

- Precise mappings cannot be created to all data sources

  Exact Mappings → ***Approximate Mappings***

- Users prefer keyword queries to structured queries

  Query Reformulation → ***Query Routing***

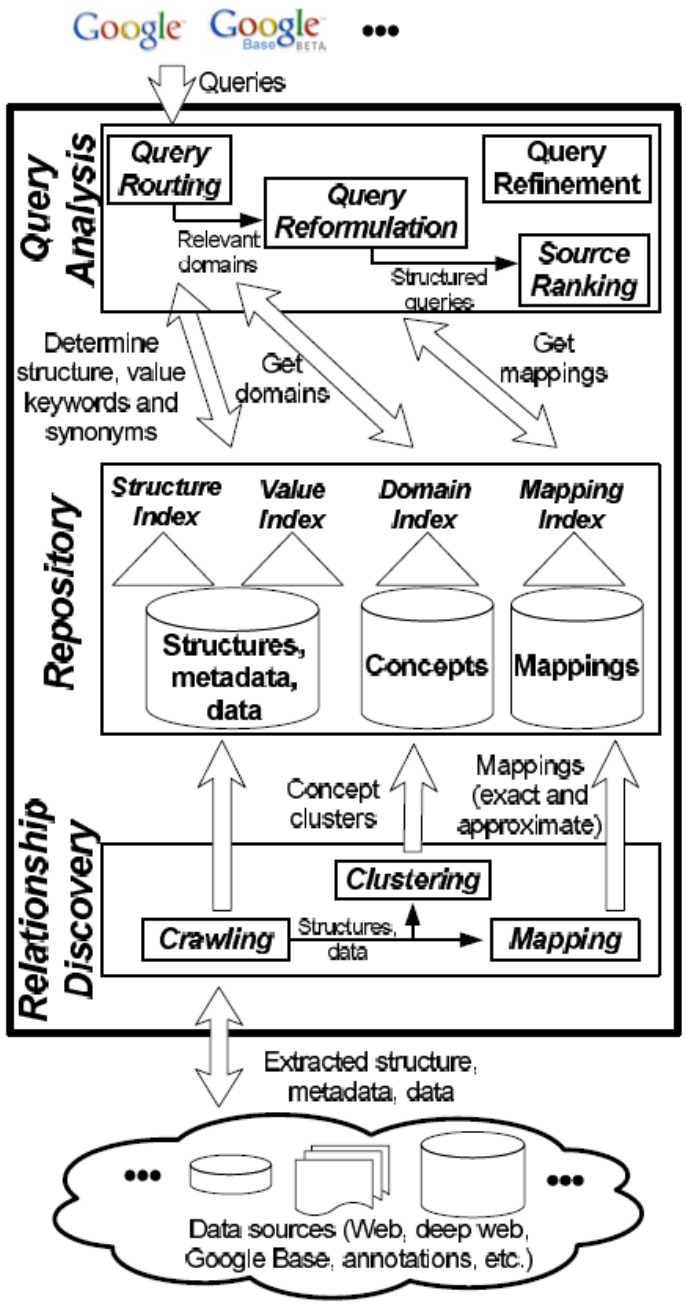
- Data sources are diverse and mappings approximate

  Exact Answers → ***Heterogeneous Result Ranking***

Uncertainty everywhere !

PAYGO Components and Principles

- *Schema clustering*
- *Approximate schema mapping*
- *Keyword queries with routing*
- *Heterogeneous result ranking*
- *Pay-as-you-go integration*
- *Modeling uncertainty at all levels*



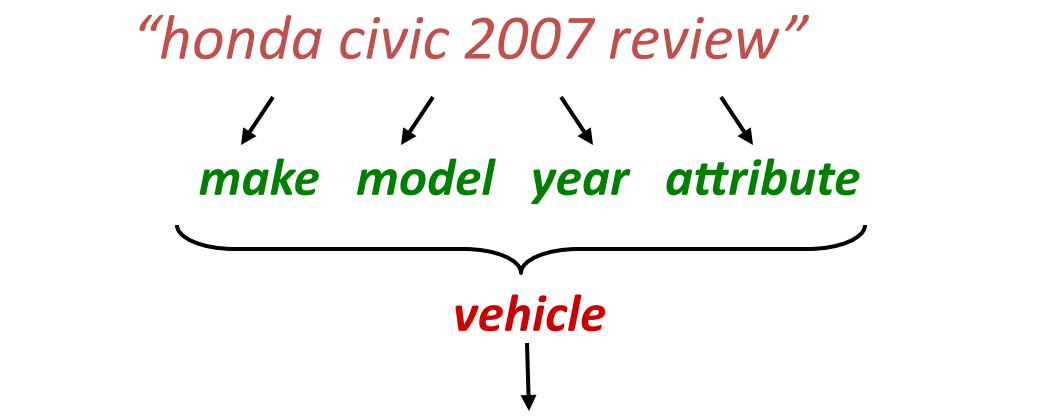
An instantiation of the PAYGO data integration architecture.

A PAYGO-based Data Integration System

- The metadata repository
- Schema clustering and mapping*(Feature Vector and Corpus based schema matching)*
- Query reformulation and answering
  - *Classify keywords*
  - *Choose domain*
  - *Generate structured queries*
  - *Rank sources*
  - *Heterogeneous Result Ranking*

Query Routing Example

- Keyword Analysis
- Domain Selection
- Query Construction
- Source Selection
- Result Ranking



car-reviews-by-year.com > car-reviews.com
> car-prices.com

Pay As You Go in PAYGO

- Integration is a *continuous* process
  - Apriori integration impossible
  - Understanding of mappings/sources/ranking/etc. evolves over time
- Mechanisms to facilitate evolution over time
  - Automatic schema clustering and matching
  - Implicit use of user feedback, e.g., from result clicks
  - Result variations to elicit disambiguating user feedback
- Queries always answered with best effort
  - "Pay" more by correcting/creating semantic mappings

Conclusion

- Web-scale Data Integration Challenge
  - Integrate large numbers of heterogeneous data sources that span many ill-defined domains
  - Support keyword queries with seamless integration of results from diverse sources

- PAYGO Architecture
  - Models uncertainty in mappings, results, and ranking
  - Evolves with time, but best effort at all times

Onto the second part !!!

Bootstrapping Pay – AS – YOU GO Data Integration

What are we going to learn in this ?

- Probabilistic Mediated Schema, How to Construct Them .
- Probabilistic Schema Mapping, How to Construct Them .
- How to automate the above two so that Data Integration can be achieved without any human effort.

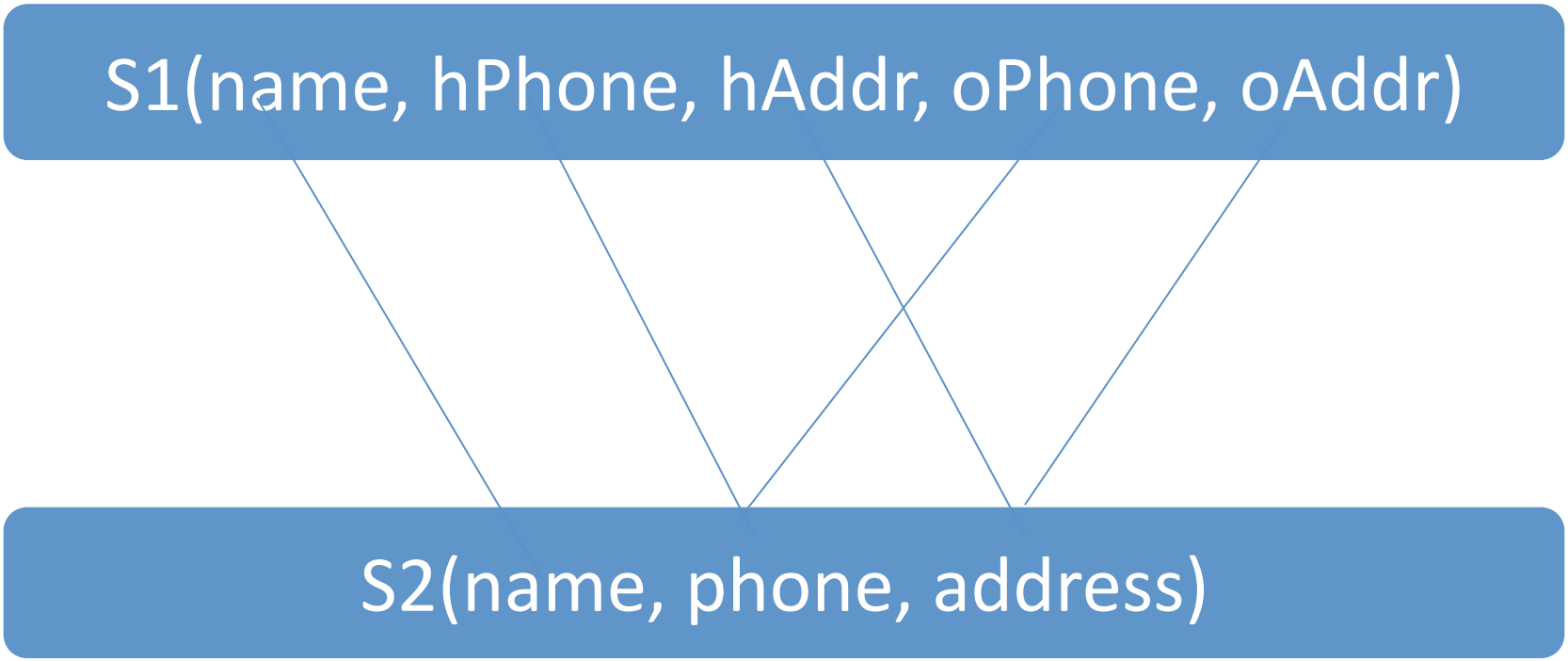
But Why Do We Need This ?

- Setting up and Maintaining DI application requires significant upfront.

- No need for full integration to start the application.

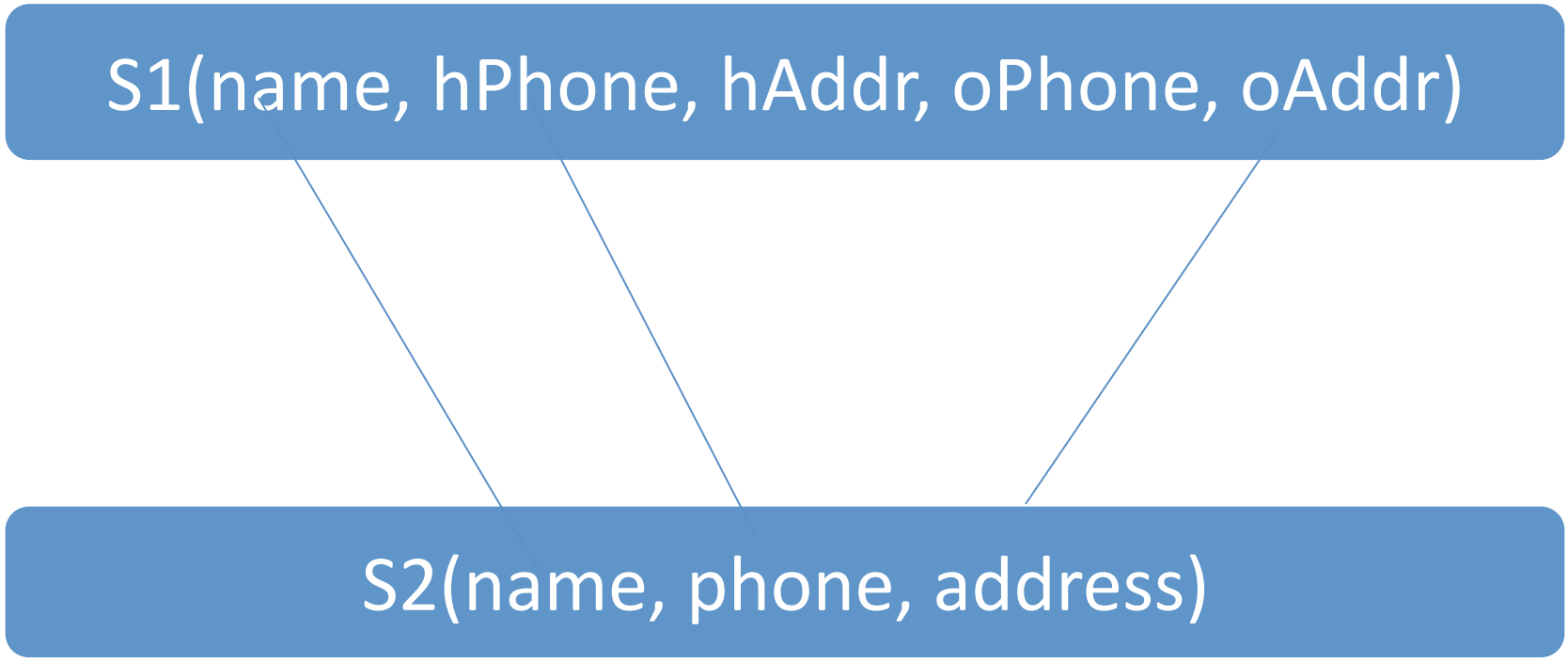
- Examples of such area include Web, Personal Information Management, Enterprise Intranets.

Example – possible clustering



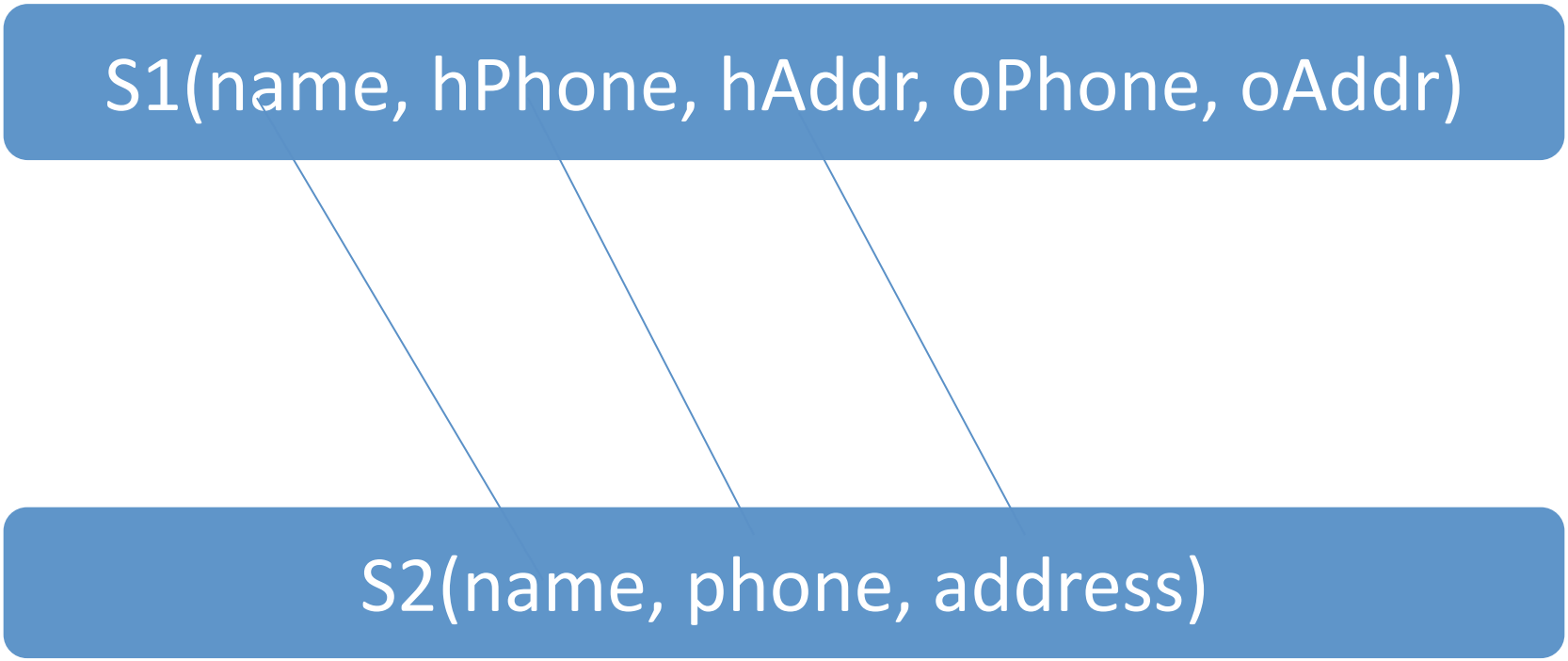
M1

Example – possible clustering

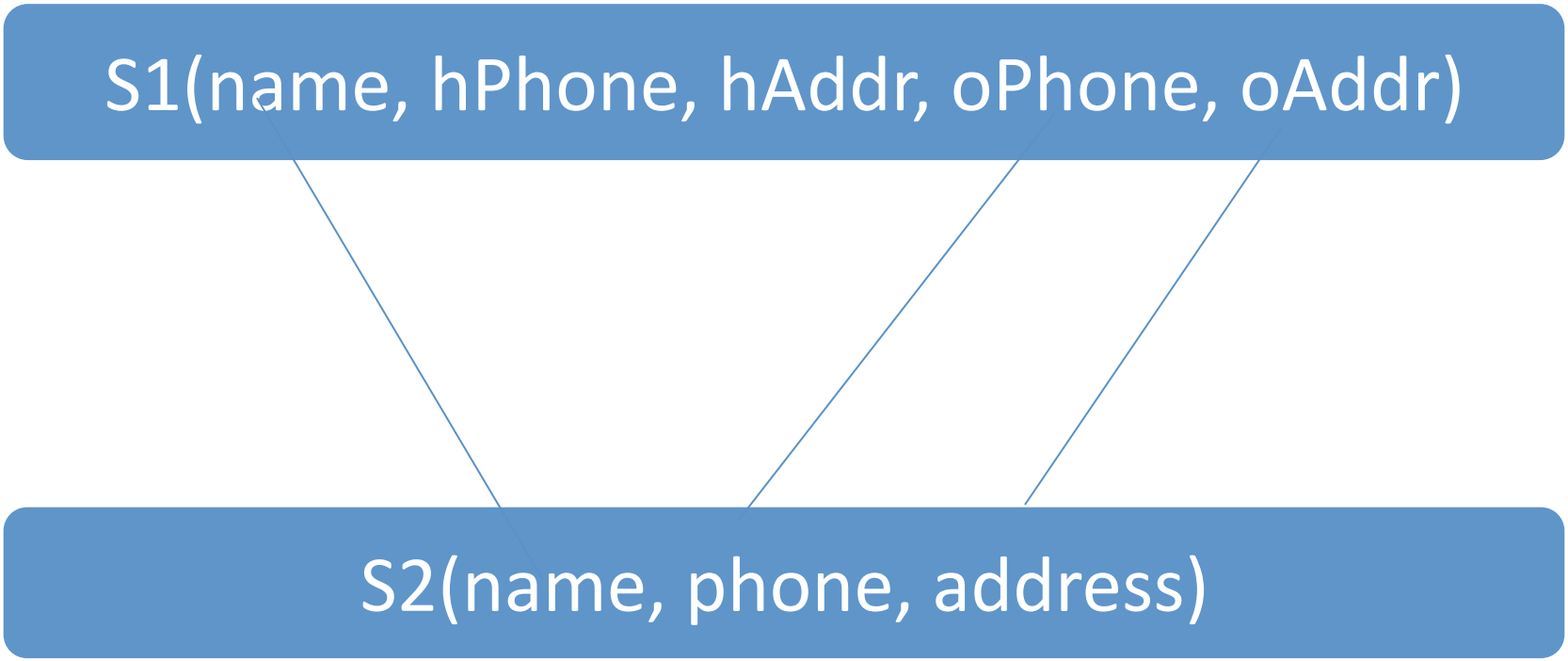


M2

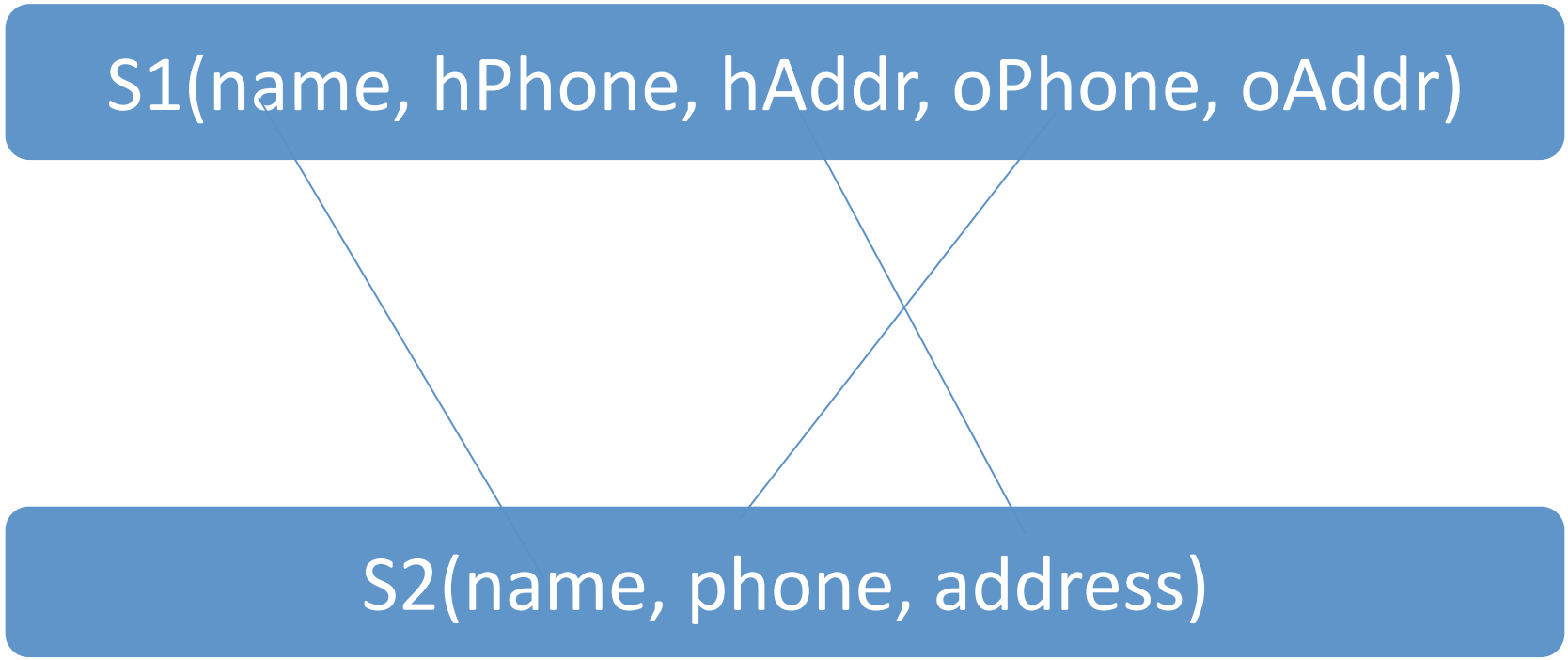
Example – possible clustering



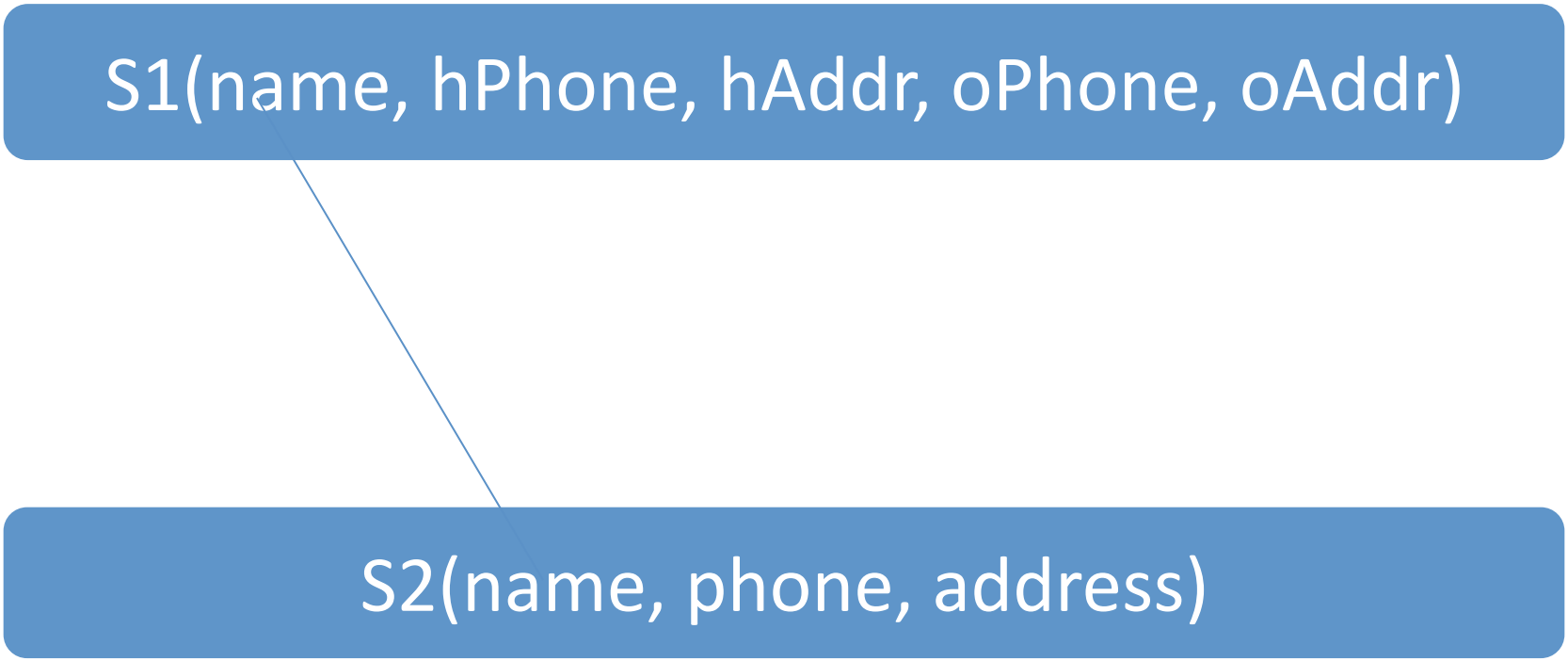
Example – possible clustering



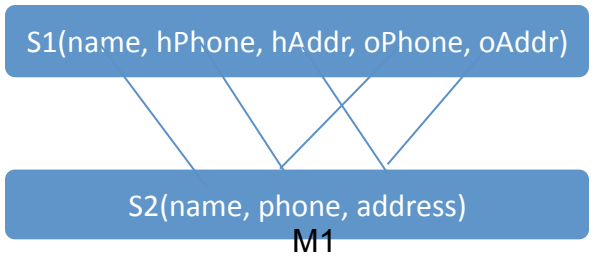
Example – possible clustering



Example – possible clustering

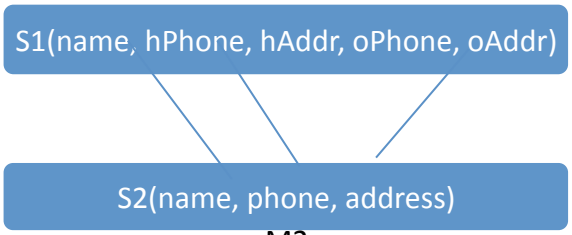


Example – possible clustering



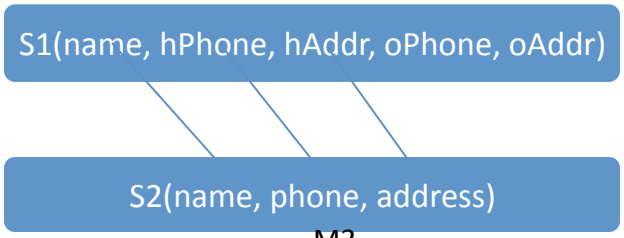
M1

Example – possible clustering



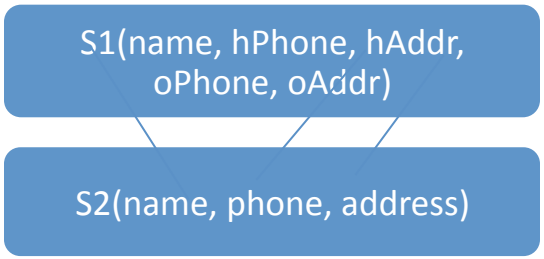
M2

Example – possible clustering



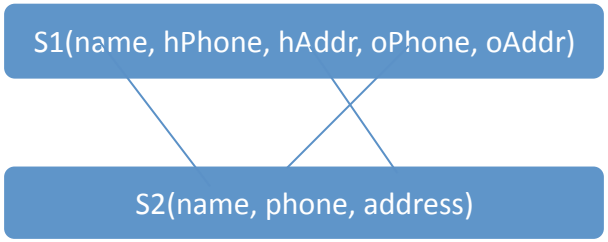
M3

Example – possible clustering



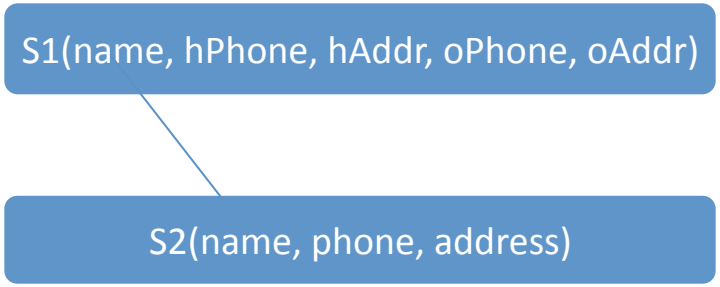
M4

Example – possible clustering



M5

Example – possible clustering



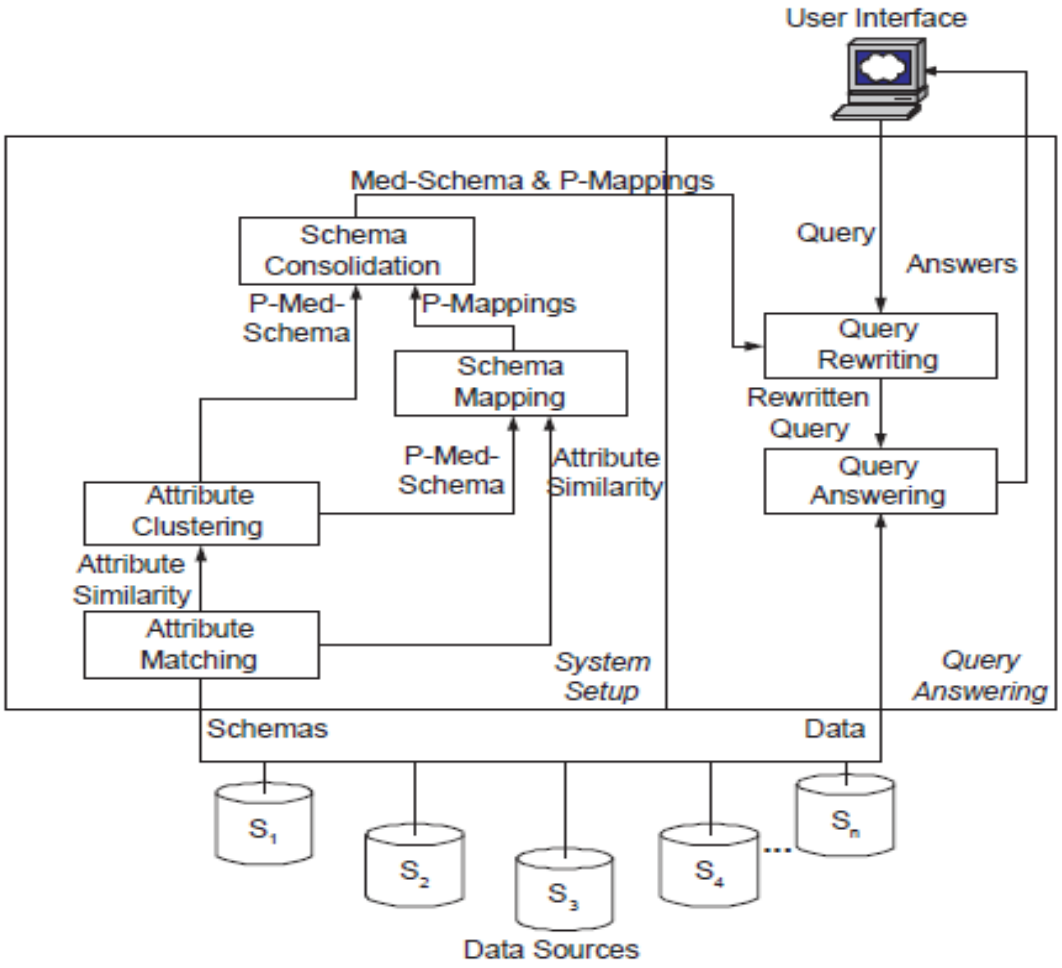
M6

- So which of these schemas should we consider?
- Even after deciding on which schema to use,

what about the mapping?

The Approach

- **Construct a probabilistic mediated schema**
- **Find best probabilistic schema mappings**
- **Create a single mediated schema to expose to the user**



The Architecture.

Example Query

- Consider the query

SELECT name, phone, address

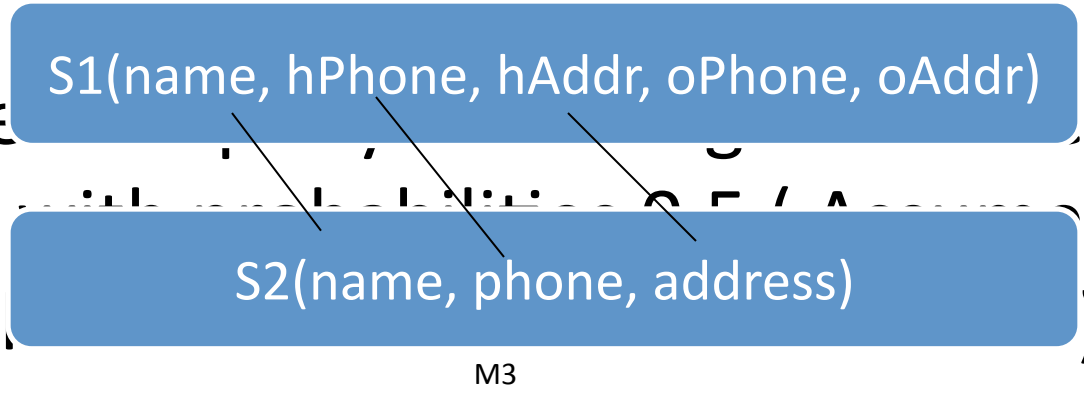
FROM People

and let

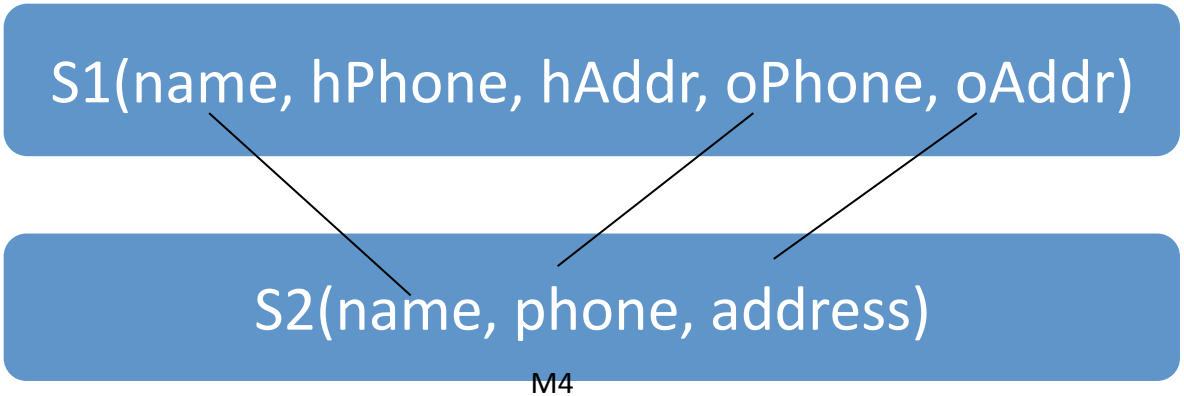
('Alice', '123-4567', '123, A Ave.', '765-4321', '456, B Ave.')

be an instance of the integrated schema

- Suppose [illegible] ma M3 and M4 [illegible] probabi[illegible])



M3



M4

- The Output of the query contains 3 fields :
  - Name
  - Phone
  - Address
- Consider the mapping

{(name, name), (hP, hPP), (oP, oP),(hA, hAA), (oA, oA)}

What does this say?

The Output

Possible Mapping	Probability
{(name, name), (hP, hPP), (oP, oP), (hA, hAA), (oA, oA)}	0.64
{(name, name), (hP, hPP), (oP, oP), (oA, hAA), (hA, oA)}	0.16
{(name, name), (oP, hPP), (hP, oP), (hA, hAA), (oA, oA)}	0.16
{(name, name), (oP, hPP), (hP, oP), (oA, hAA), (hA, oA)}	0.04

(a)

Possible Mapping	Probability
{(name, name), (oP, oPP), (hP, hP), (oA, oAA), (hA, hA)}	0.64
{(name, name), (oP, oPP), (hP, hP), (hA, oAA), (oA, hA)}	0.16
{(name, name), (hP, oPP), (oP, hP), (oA, oAA), (hA, hA)}	0.16
{(name, name), (hP, oPP), (oP, hP), (hA, oAA), (oA, hA)}	0.04

(b)

Answer	Probability
('Alice', '123-4567', '123, A Ave.')	0.34
('Alice', '765-4321', '456, B Ave.')	0.34
('Alice', '765-4321', '123, A Ave.')	0.16
('Alice', '123-4567', '456, B Ave.')	0.16

(c)

P-Mediated Schema

- *Let* $\{S_1, \ldots, S_n\}$ *be a set of schemas.*

- *A probabilistic mediated schema* (p-med-schema) for $\{S_1, \ldots, S_n\}$ *is a set* $M = \{(M_1, Pr(M_1)), \ldots, (M_l, Pr(M_l))\}$

P-Mapping

- *Let S be a source schema and M be a mediated schema.*
- *A probabilistic schema* mapping (p-mapping) *between S and M is a set*

  $pM = \{(m_1, Pr(m_1)), \ldots, (m_l, Pr(m_l))\}$
- The focus is on one to one mapping but one to many mapping is possible.

Semantics of Queries

- Importance is on Top – k precision
-

In short: probability of tuple as an output is the summation of all the probabilities

Le

p *s*

wh

an

Let t be a tuple. Let Pr(t|Mi)) i=[1, l], be the probability of t in the answer of Q with respect to Mi and pM(Mi). Let $p = l_i=1Pr(t|M_i) * Pr(M_i)$. *If p > 0, then we say (t, p) is a by-table answer with respect to Mand pM.*

We denote all by-table answers by QM,pM(D).

Probabilistic Mediated schemas VS Deterministic Mediated Schema

- Now consider one-to-many schema mappings, *Given a source schema S, a p-med-schema M, and a set of p-mappings pM between S and possible mediated schemas in M, there exists a deterministic mediated schema T and a p-mapping* pM *between S and T, such that* any combination of a p-med-schema and p-mappings can be equivalently represented using a deterministic mediated schema with p-mappings,

Too Complicated?

But

- *There e* pM *between S and T, and an instance D of S, such that for any p-med-schema M and any set m of deterministic mappings between S and possible mediated schemas in M, there exists a query Q such that* $Q_{M,m}(D) \neq Q_{T,pM}(D)$.

may not be represented using a p-med-schema with deterministic schema mappings.

Now Consider This Statement

- If we restrict our attention to one-to-one mappings then a probabilistic mediated schema does add expressive power.

Theorem: There exists a source schema S, a p-med-schema M, a set of one-to-one p-mappings pM between S and M, and a query Q over M, such that for any deterministic one-to-one p-mapping pM between S and T, there exists a query Q such that, $Q_{M,pM}(D) \neq Q_{T,pM}(D)$

Rephrase

Conclusion

- Constructing one-to-many p-mappings in practice is much harder than constructing one-to-one p-mappings.

- When we are restricted to one-to-one p-mappings, p-med-schemas grant us more expressive power while keeping the process of mapping generation feasible.

Creating Single Mediated Schema

- Remove Infrequent Attributes.
- Construct Weighted Graph (Threshold τ).
- Cluster the nodes in the resulting weighted graph to obtain the mediated schema.

Creating a p-med-schema

S1: (name,address,email-address)

S2: (name,home-address)

- 0: **Input:** Source schemas $S_1, \ldots, S_n$.

  Output: A set of possible mediated schemas.
- 1: Compute $A = \{a_1, \ldots, a_m\}$, the set of all source attributes;
- 2: **for each** ($j \in [1,m]$)

  Compute frequency $f(a_j) = \frac{|\{i \in [1,n] \mid a_j \in S\}|}{n}$
- 3: Set $A = \{a_j \mid j \in [1,m], f(a_j) \geq \theta\}$; //$\theta$ *is a threshold*
- 4: Construct a weighted graph G(V,E), where

  (1) V = A, and

  (2) for each $a_j, a_k \in A$, $s(a_j, a_k) \geq \tau - \varrho$, there is an edge
- (a_j, a_k) with weight $s(a_j, a_k)$;
- 5: Mark all edges with weight less than $\tau + \varrho$ as *uncertain;*
- 6: **for each** (uncertain edge $e = (a_1, a_2) \in E$)

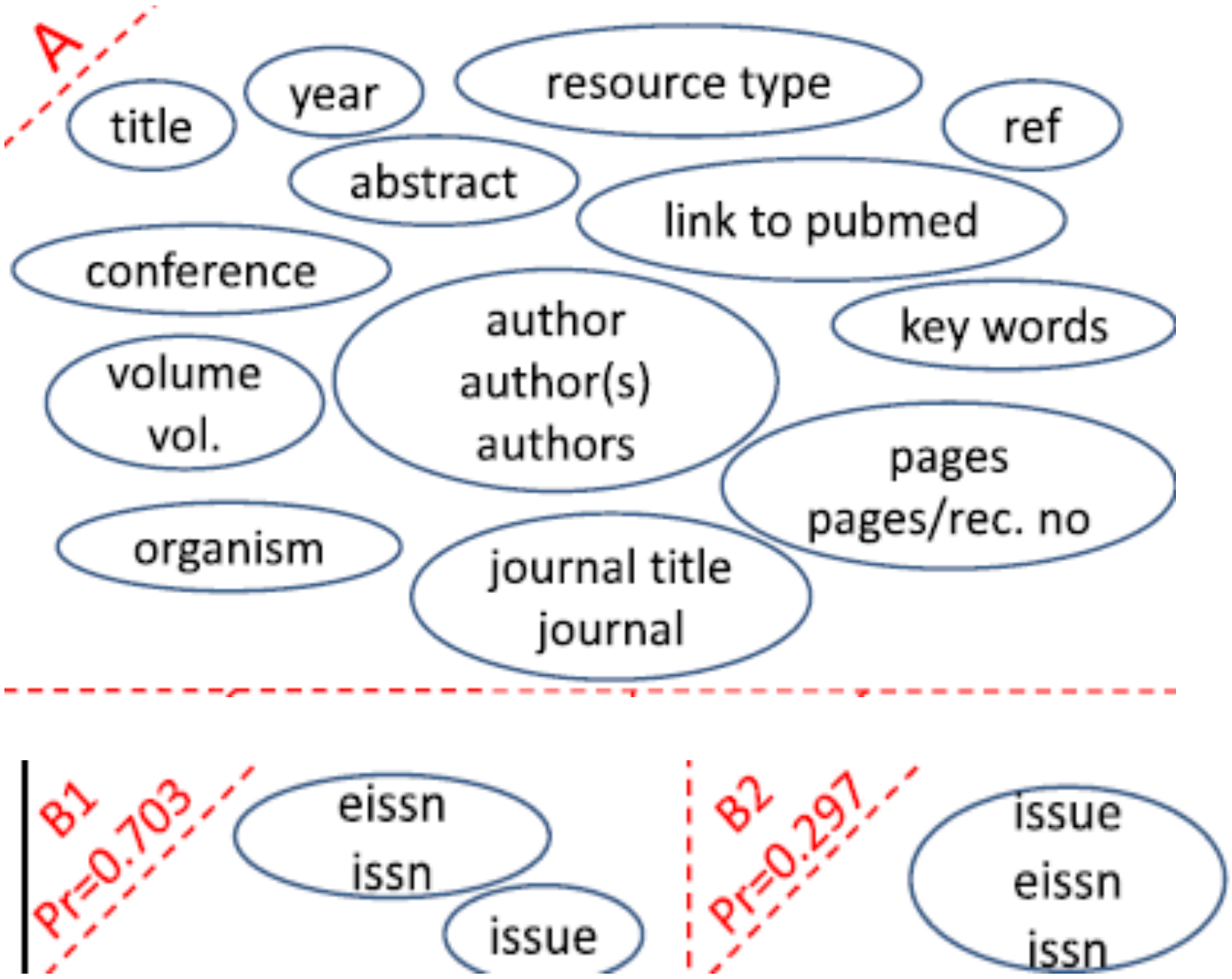
  Remove e from E if (1) a_1 and a_2 are connected by a path with only certain edges, or (2) there exists $a3 \in V$, such that a2 and a3 are connected by a path with only certain edges and there is an uncertain edge (a1, a3);
- 7: **for each** (subset of uncertain edges)

  Omit the edges in the subset and compute a mediated schema where each connected component in the graph corresponds to an attribute in the schema;
- 8: **return** distinct mediated schemas.

Consistency

- *Let M be a mediated schema for sources S_1. . . . . S_n. We say M is consistent with a source schema S_i, $i \in$ [1, n' ... appear in th...*

mediated schema is consistent with a source only if it does not group distinct attributes in the source


A
title
year
resource type
ref
abstract
link to pubmed
conference
author
author(s)
authors
key words
volume
vol.
pages
pages/rec. no
organism
journal title
journal
B1
Pr=0.703
eissn
issn
issue
B2
Pr=0.297
issue
eissn
issn

Weighted Correspondence

- It specifies the degree of similarity between a pair of attributes
- Formula: $$p_{i,j} = \sum_{a \in A_j} s(a_i, a).$$

Generating p – mapping Example

- pM1:

m1: (A,A'), (B,B'): 0.3

m2: (A,A'): 0.3

m3: (B,B'): 0.2

m4: empty: 0.2

- pM2:

m1: (A,A'), (B,B'): 0.5

m2: (A,A'): 0.1

m3: empty: 0.4

Generating p – mapping

- Enumerate all possible one-to-one schema mappings between S and M that contain a subset of correspondences in C.

- We assign probabilities on each of the mappings in a way that maximizes the entropy of our result p-mapping.

Consolidating the Schemas

Advantages:

- The user expects to see a single schema
- Queries now need to be rewritten and answered based on only one mediated schema

Requirements:

- The answers to queries over the consolidated schema be equivalent to the ones over the probabilistic mediated schema.

The Algorithm

- 0: **Input: Mediated schemas M1, . . . ,Ml.**

  Output: A consolidated single mediated schema T.
- 1: Set T =M1.
- 2: **for (i = 2, . . . , l) modify T as follows:**
- 3: **for each (attribute A′ in Mi)**
- 4: **for each (attribute A in T)**
- 5: Divide A into A ∩ A′ and A − A′ ;
- 6: **return T.**
- *Consider a p-med-schema M = {M1,M2}, where M1 contains three attributes {a1, a2, a3}, {a4}, and* {a5, a6}*, and M2 contains two attributes {a2, a3, a4} and* {a1, a5, a6}*. The target schema T would then contain four attributes:* {a1}*, {a2, a3}, {a4}, and {a5, a6}.*

Consolidating the p-mapping

- Update the Correspondence
- Update the probabilities.(note: sum may not be 1)
- Consolidate.

- Finally by theorem of Merge Equivalence we conclude that *For all queries Q, the answers obtained by posing Q over a p-med-schema M =*{M1, . . . ,Ml} *with p-mappings pM1, . . . , pMl is equal to the answers obtained by posing Q over the consolidated mediated schema* T *with consolidated p-mapping pM.*

Experiments

- Setup:
  - UDI accepts select – project queries and returns ranked output based on the their probabilities.
  - Mediated schema has only one table and so no join.
- UDI transforms it into a set of queries on the data sources according to the probabilistic schema mappings, retrieves answers from individual data sources, and then combines the answers assuming that the data sources are independent

Setup Continued

- Tools used
  - MySQL – To store date
  - SecondString - Jaro Winkler Similarity
  - Knitro – Maximizing entropy in p-mapping.
  - Windows Vista machine with Intel Core 2 GHz CPU and 2GB memory.
- Thresholds:
  - Similarity threshold : 0.85
  - Error bar for uncertainty : 0.02
  - Attributes in mediated schema – 10%
  - Correspondence threshold - 0.85

Data and Queries

- Chose 5 domains
- Each Table 10 – 100 tuples.
- 10 queries; 4 attributes in select; 0 – 3 in where
- Allowed Operators $\neq$ =,<,≤,>,≥ and LIKE.

Performance Measures.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Precision, recall and F-measure of query answering of the UDI system compared with a manually created integration system. The results show that UDI obtained a high accuracy in query answering.

Domain	Precision	Recall	F-measure
Golden standard			
People	1	.849	.918
Bib	1	.852	.92
Approximate golden standard			
Movie	.95	1	.924
Car	1	.917	.957
Course	.958	.984	.971
People	1	1	1
Bib	1	.955	.977

Results

- Obtained a recall of about 0.85 on the two domains
- In comparison to the approximate golden standard, we obtained a recall of over 0.9 in all cases and over 0.95 in four of the domains
- Extrapolating from the discrepancy expected recall would be around 0.8-0.85 with respect to the golden standard on all domains.

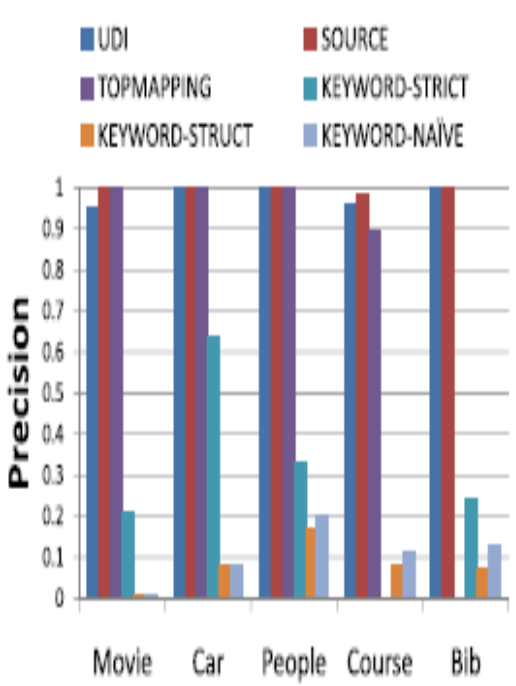
Scope to improve?

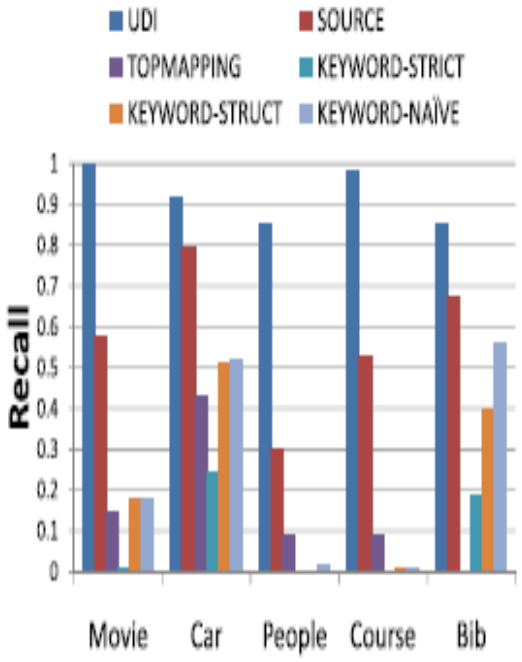
- Yes, matcher considered only similarity of attribute names.
- Did not look at values in the corresponding columns or other clues
- Eg. Location and address
- suffered some loss of recall because we set a high threshold to choose attribute correspondences in order to reduce the number of correspondences considered in the entropy maximization

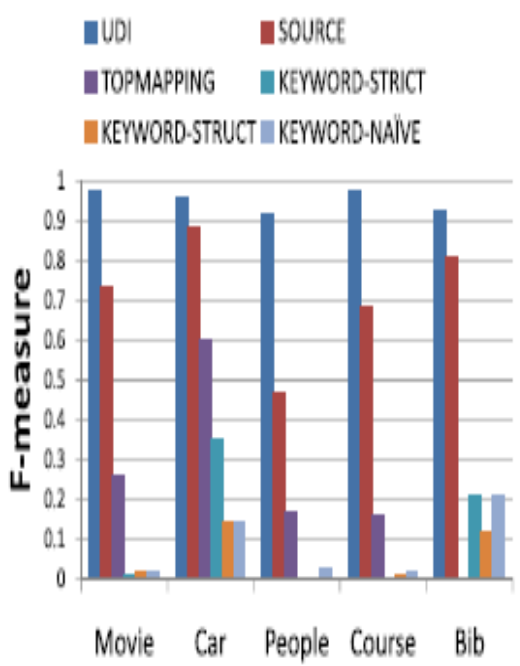
Competing automatic approaches

- The first approach is to consider the data sources as a collection of documents and perform keyword search.
  - KEYWORDNAIVE
  - KEYWORDSTRUCT
  - KEYWORDSTRICT
- SOURCE, answers Q directly on every data source that contains all the attributes in Q, and takes the union of returned answers
- TOPMAPPING approach

Result

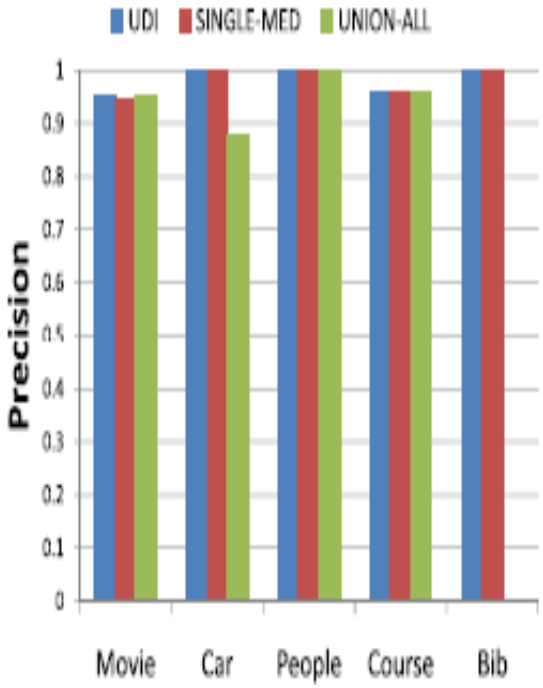


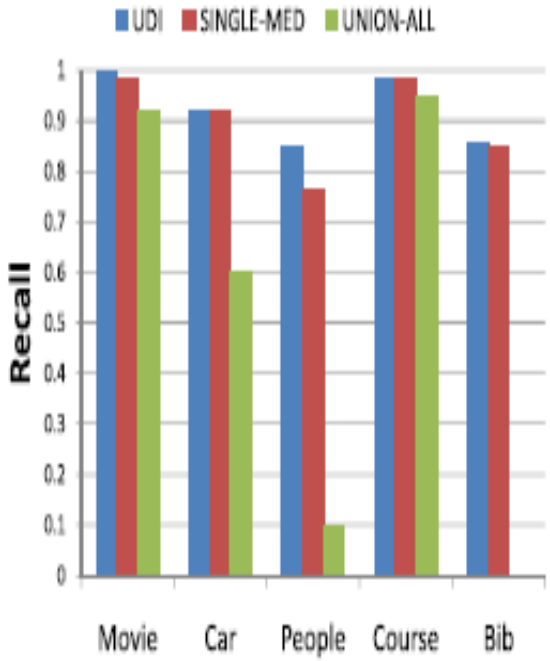


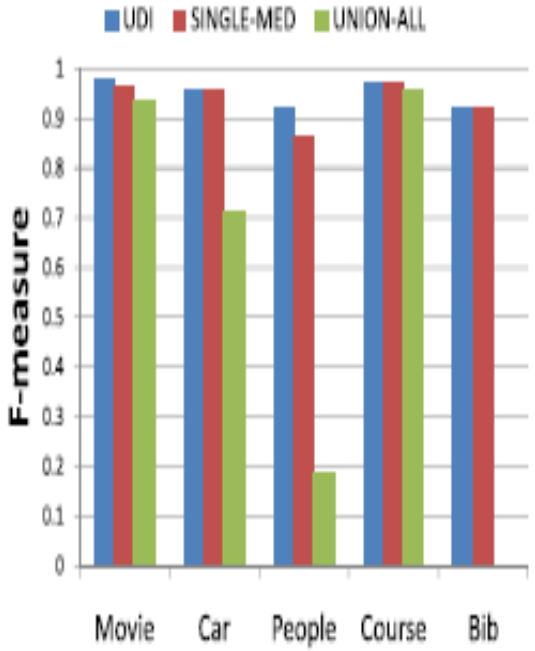


Performance of query answering of the UDI system and alternative approaches. The UDI system obtained the highest F-measure in all domains.

Contribution of p-med-schema





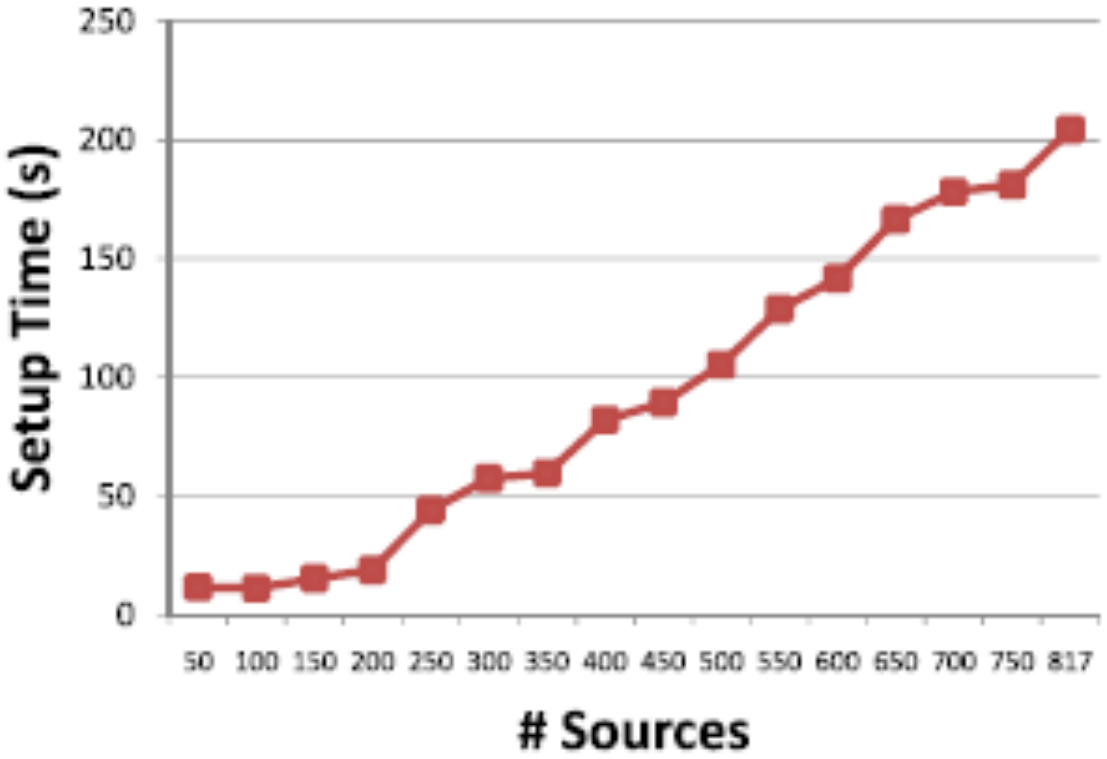


Performance of query answering of the UDI system and approaches that generate deterministic mediated schemas. The experimental results show that using a probabilistic mediated schema improves query answering performance. Note that we did not plot the measures for UNIONALL in the Bib domain as this approach ran out of memory in system setup.

Precision, recall and F-measure of p-med-schemas generated by UDI

Domain	Precision	Recall	F-measure
Movie	.97	.62	.76
Car	.68	.88	.77
People	.76	.86	.81
Course	.83	.58	.68
Bib	.77	.81	.79
Avg	.802	.75	.762

Setup efficiency



System setup time for the Car domain. When the number of data sources was increased, the setup time increased linearly

Related Works

- He and Chang - approach was to create a mediated schema that is statistically maximally *consistent with the source schemas*
- Magnani et al. [20] proposed generating a set of alternative mediated schemas based on probabilistic relationships between *relations*
- Dong et al. [10] proposed the concept of probabilistic schema mapping and studied query answering with respect to such mappings, but they did not describe how to create such mappings.
- Magnani and Montesi [19] have empirically shown that top-k schema mappings can be used to increase the recall of a data integration process and Gal [13] described how to generate top-k schema matching by combining the matching results generated by various matchers.

Conclusion

- Possible to automatically set up a data integration application that obtains answers with high precision and recall.
- main novel element we introduced to build our system is a probabilistic mediated schema, which is constructed automatically by analyzing the source schemas

Conclusion – Future Work

- How to improve the data integration system with time.
- Pinpoint where human feedback can be most effective in improving the semantic integration in the system
- Extend the techniques to dealing with multiple-table sources, including mapping multi-table schemas, normalizing mediated schemas, and so on.

Acknowledgements

- Some of the slides have been adapted from presentation by the authors of the paper.

  http://www.cidrdb.org/cidr2007/slides/p40-madhavan.*ppt*
- Contents have been referred from websites like en.wikipedia.org