Information Integration on the Web

AAAI Tutorial (SA2)

Monday July 22nd 2007. 9am-1pm

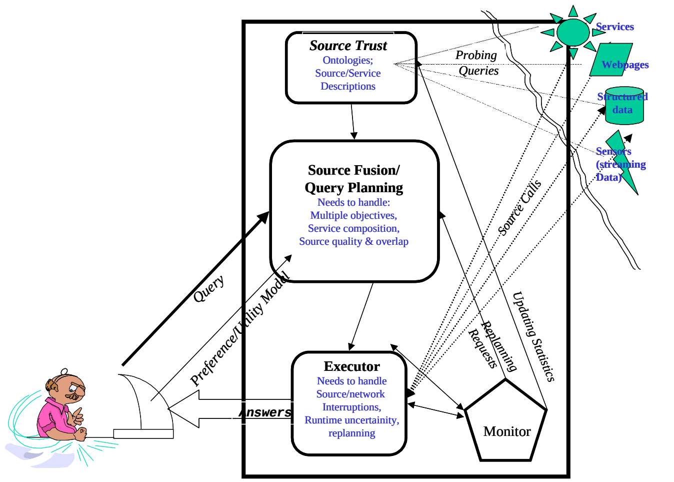
Rao Kambhampati & Craig Knoblock

ARIZONA STATE UNIVERSITY

USC UNIVERSITY OF SOUTHERN CALIFORNIA

Overview

- Motivation & Models for Information Integration [30]
- Getting Data into structured format [30]
- Getting Sources into alignment [30]
- Getting Data into alignment [30]
- Processing Queries [45]
- Wrapup [15]



Preamble & Platitudes

- Internet is growing at an *ginormous* rate
- All kinds of information sources are *online*
  - Web pages, Data Sources (~25M), Sensors, Services
- Promise of unprecedented information access to lay public
  - *But, right now, they still need to "know" where to go, and be willing to manually put together bits and pieces of information gleaned from various sources and services*

"Information Integration" aims to do this automatically.

Combining information from multiple autonomous information sources, and answering queries using the combined information

Information Integration

- Combining information from multiple autonomous information sources
  - And answering queries using the combined information
- Many Applications
  - WWW:
    - Comparison shopping
    - Portals integrating data from multiple sources
    - B2B, electronic marketplaces
    - Mashups, service composion
  - Science informatics
    - Integrating genomic data, geographic data, archaeological data, astro-physical data etc.
  - Enterprise data integration
    - An average company has 49 different databases and spends 35% of its IT dollars on integration efforts

Blind Men & the Elephant: Differing views on Information Integration

Database View

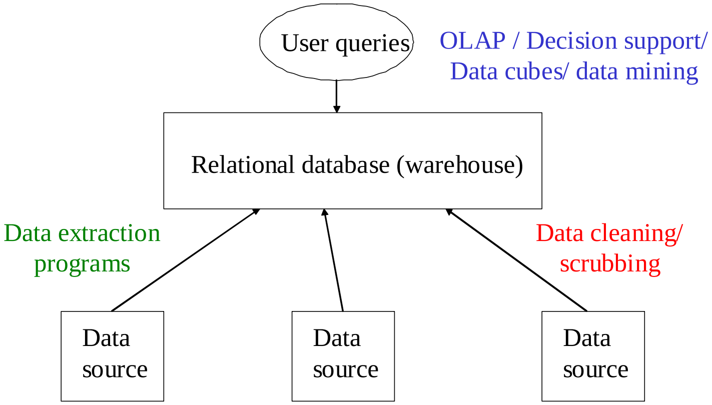
- Integration of autonomous structured data sources
- Challenges: Schema mapping, query reformulation, query processing

Web service view

- Combining/composing information provided by multiple web-sources
- Challenges: learning source descriptions; source mapping, record linkage etc.

IR/NLP view

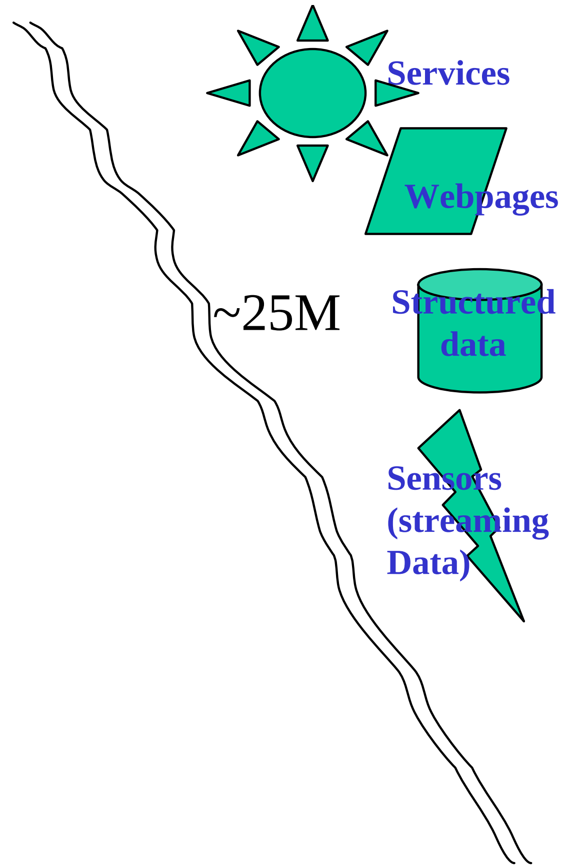
- Computing textual entailment from the information in disparate web/text sources
- Challenges: Convert to structured format

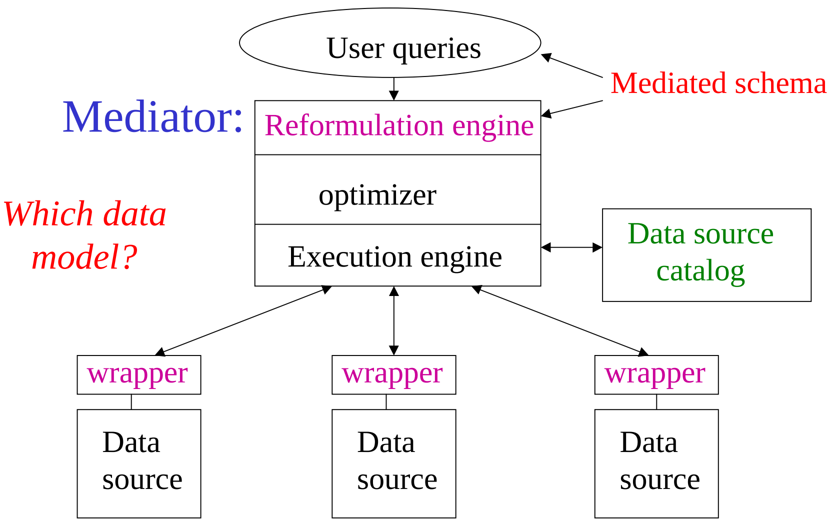


--Warehouse Model--

--Get all the data and put into a local DB

--Support structured queries on the warehouse





--Mediator Model--

--Design a mediator

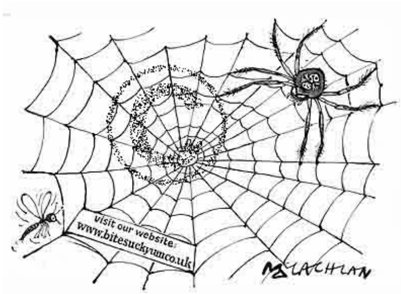
--Reformulate queries

--Access sources

--Collate results

Challenges

- Extracting Information
- Aligning Sources
- Aligning Data
- Query reformulation
- Indexing with Structure



--Search Model--

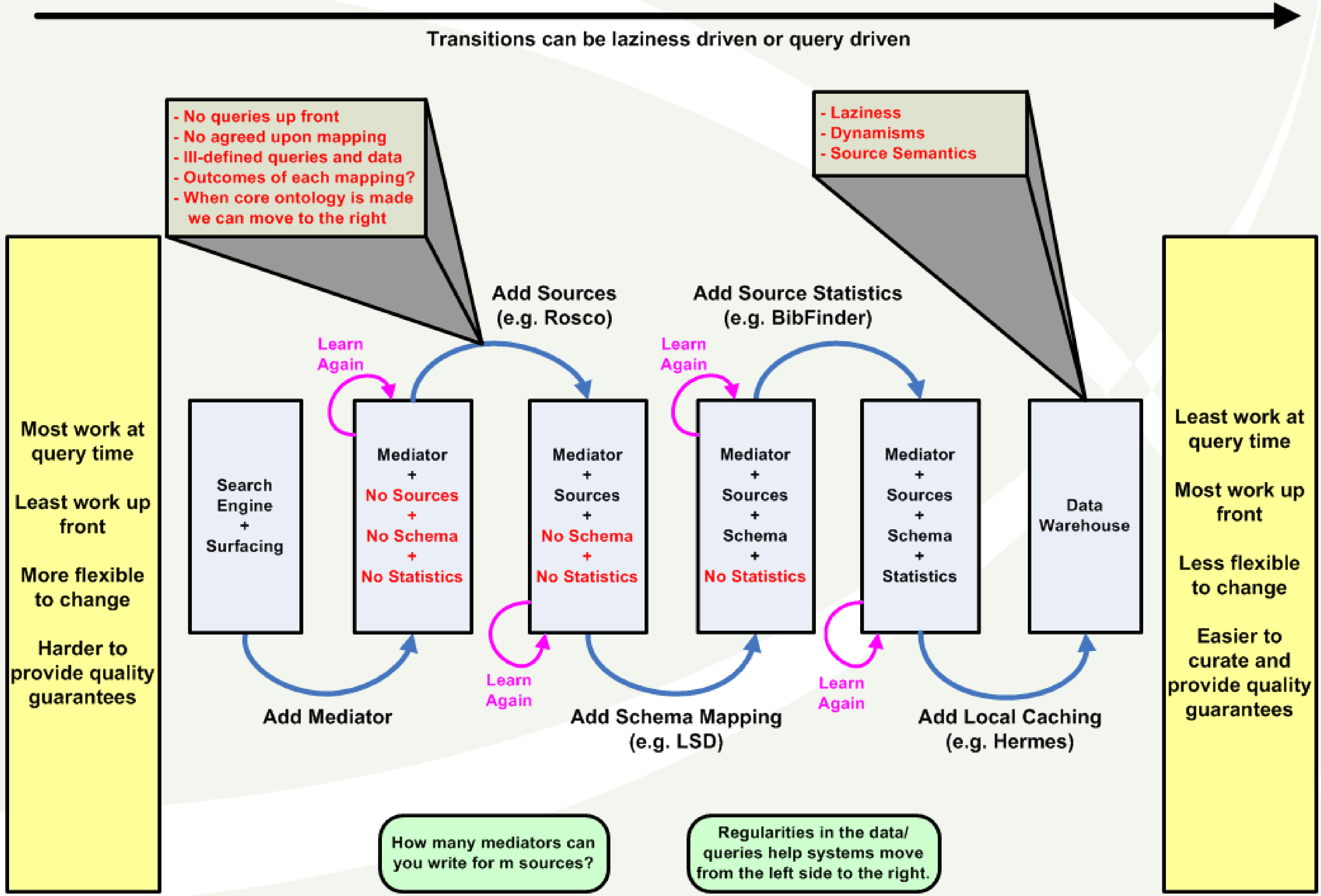
--Materialize the pages

--crawl &index them

--include them in search results

Mediator Systems

Moving from Ad hoc Integration to Data Warehouses



Dimensions of Variation

- Conceptualization of (and approaches to) information integration vary widely based on
  - Type of data sources: being integrated (text; structured; images etc.)
  - Type of integration: vertical *vs*. horizontal *vs*. both
  - Level of up-front work: *Ad hoc* vs. pre-orchestrated
  - Control over sources: Cooperative sources vs. Autonomous sources
  - Type of output: Giving answers vs. Giving pointers
  - Generality of Solution: Task-specific (Mashups) vs. Task-independent (Mediator architectures)

Dimensions: Type of Data Sources

- Data sources can be
  - Structured (e.g. relational data)
  - Text oriented
  - Multi-media (e.g. images, maps)
  - Mixed

No need for information extraction

Dimensions: Vertical vs. Horizontal

- Vertical: Sources being integrated are all exporting same type of information. The objective is to collate their results
  - Eg. Meta-search engines, comparison shopping, bibliographic search *etc.*
  - Challenges: Handling overlap, duplicate detection, source selection
- Horizontal: Sources being integrated are exporting different types of information
  - E.g. Composed services, Mashups,
  - Challenges: Handling "joins"
- Both..

Dimensions: Level of Up-front work
Ad hoc vs. Pre-orchestrated

- Fully Query-time II (blue sky for now)
  - Get a query from the user on the mediator schema
  - Go "discover" relevant data sources
  - Figure out their "schemas"
  - Map the schemas on to the mediator schema
  - Reformulate the user query into data source queries
  - Optimize and execute the queries
  - Return the answers

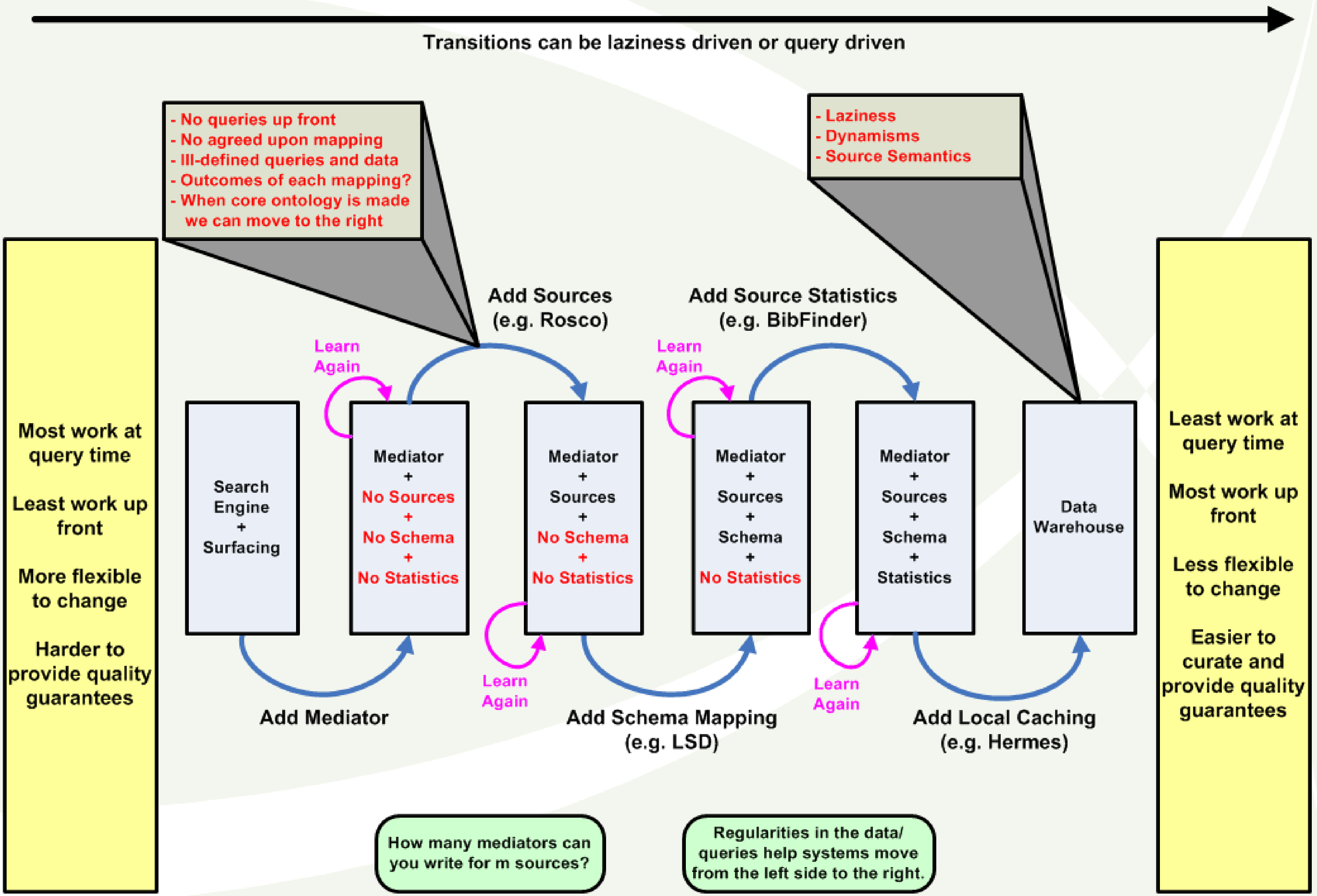
(most interesting action is "in between")

E.g. We may start with known sources and their known schemas, do hand-mapping and support automated reformulation and optimization

- Fully pre-fixed II
  - Decide on the only query you want to support
  - Write a (java)script that supports the query by accessing specific (pre-determined) sources, piping results (through known APIs) to specific other sources
    - Examples include Google Map Mashups

Mediator Systems

Moving from Ad hoc Integration to Data Warehouses



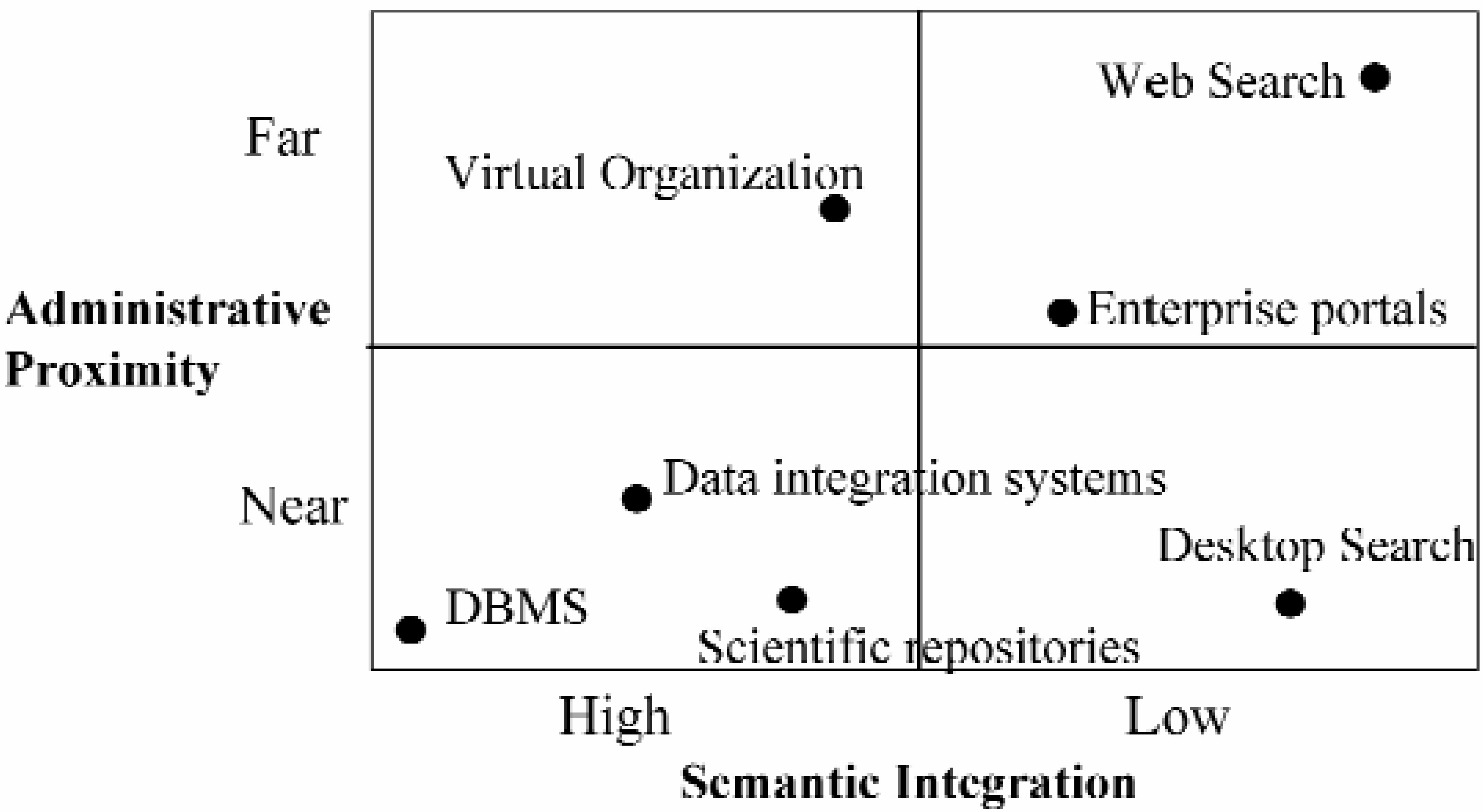
Dimensions: Control over Sources (Cooperative vs. Autonomous)

- Cooperative sources can (depending on their level of kindness)
  - Export meta-data (e.g. schema) information
  - Provide mappings between their meta-data and other ontologies
    - Could be done with Semantic Web standards…
  - Provide unrestricted access
    - Examples: Distributed databases; Sources following semantic web standards
- …for uncooperative sources all this information has to be gathered by the mediator
    - Examples: Most current integration scenarios on the web

Dimensions: Type of Output (Pointers vs. Answers)

- The cost-effective approach may depend on the quality guarantees we would want to give.
- At one extreme, it is possible to take a "web search" perspective—provide potential answer pointers to keyword queries
  - Materialize the data records in the sources as HTML pages and add them to the index
    - Give it a sexy name: Surfacing the deep web
- At the other, it is possible to take a "database/knowledge base" perspective
  - View the individual records in the data sources as assertions in a knowledge base and support inference over the entire knowledge.

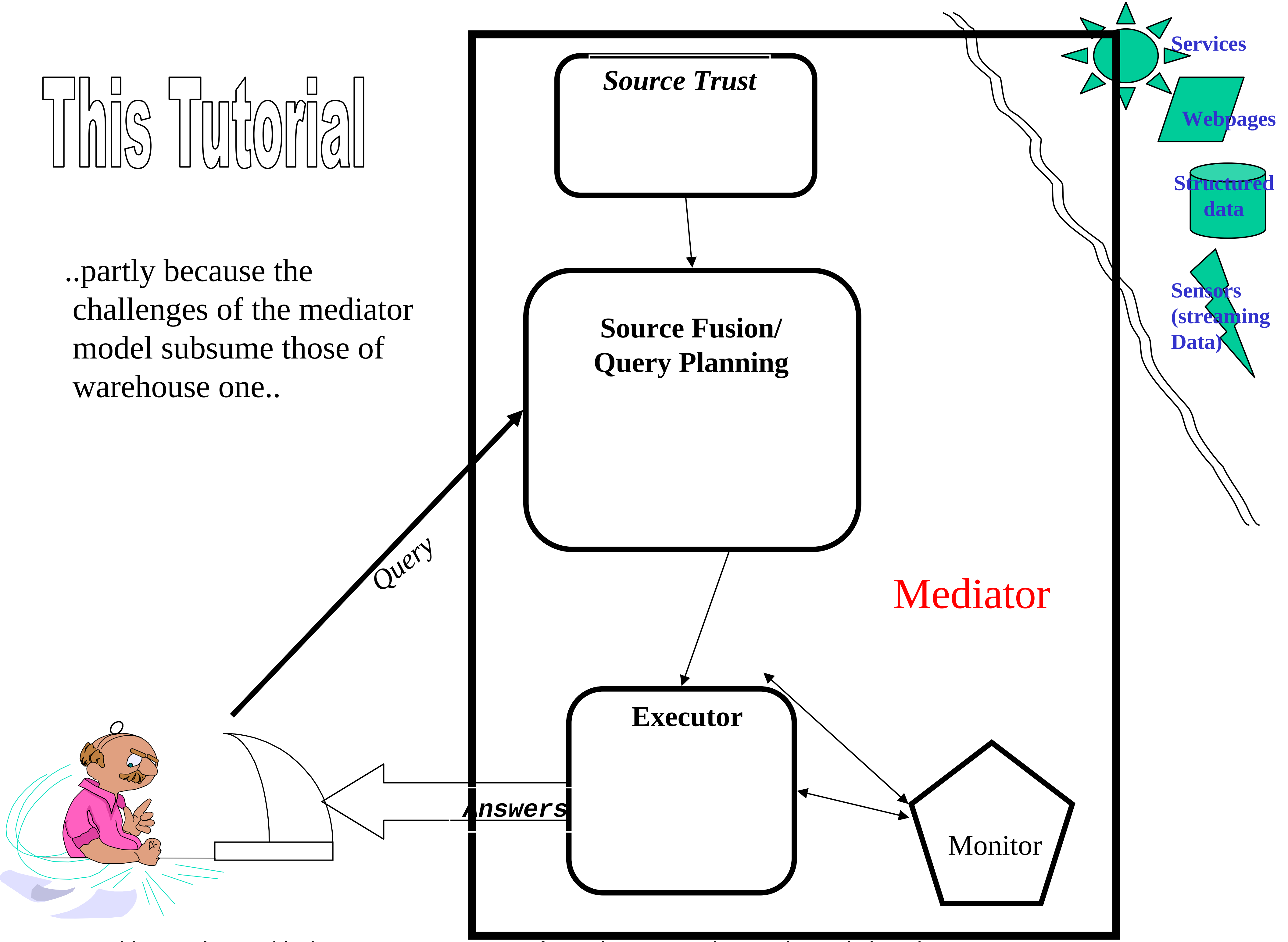
Interacting Dimensions..



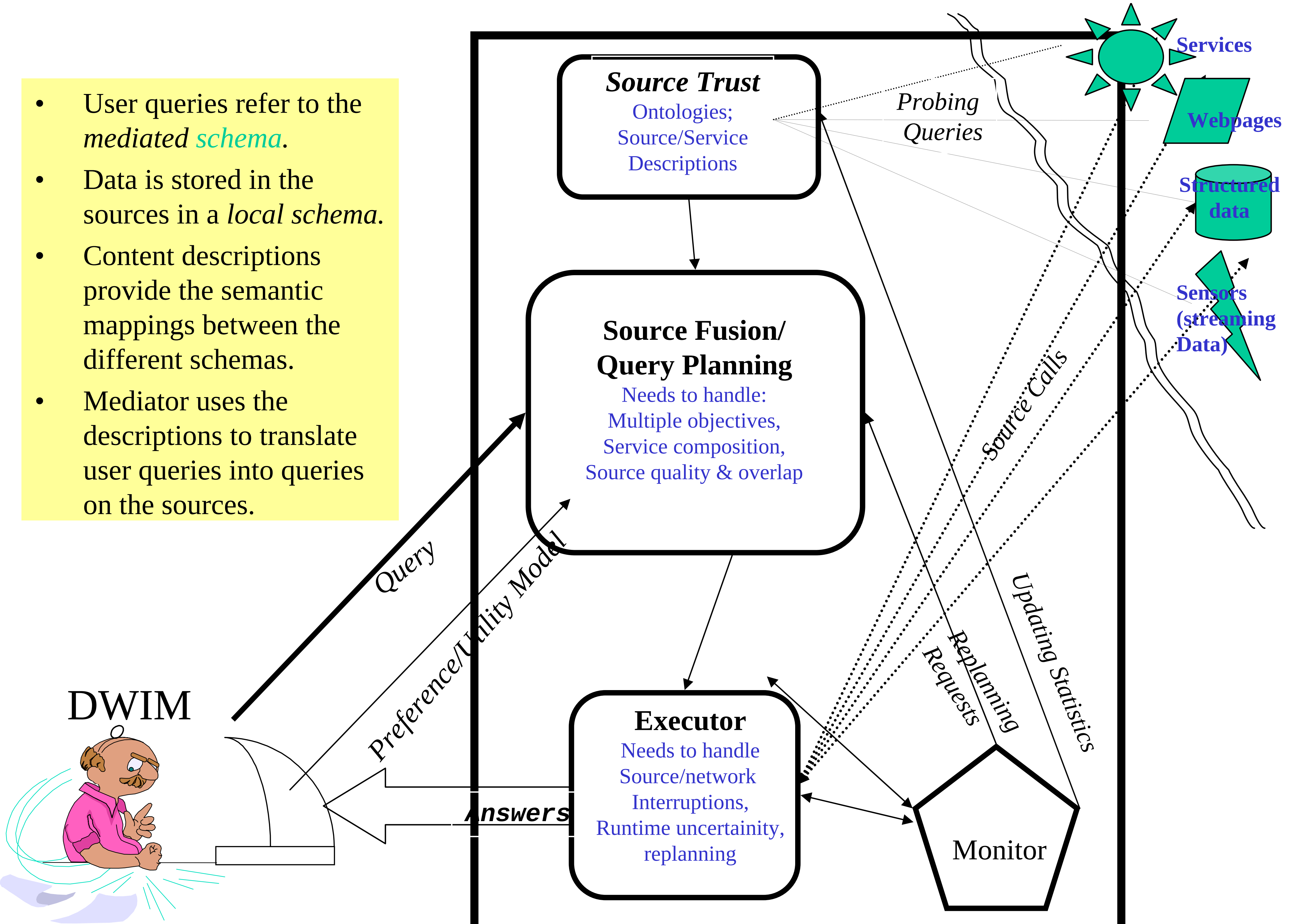
[Figure courtesy Halevy et. Al.]

This Tutorial

..partly because the challenges of the mediator model subsume those of warehouse one..

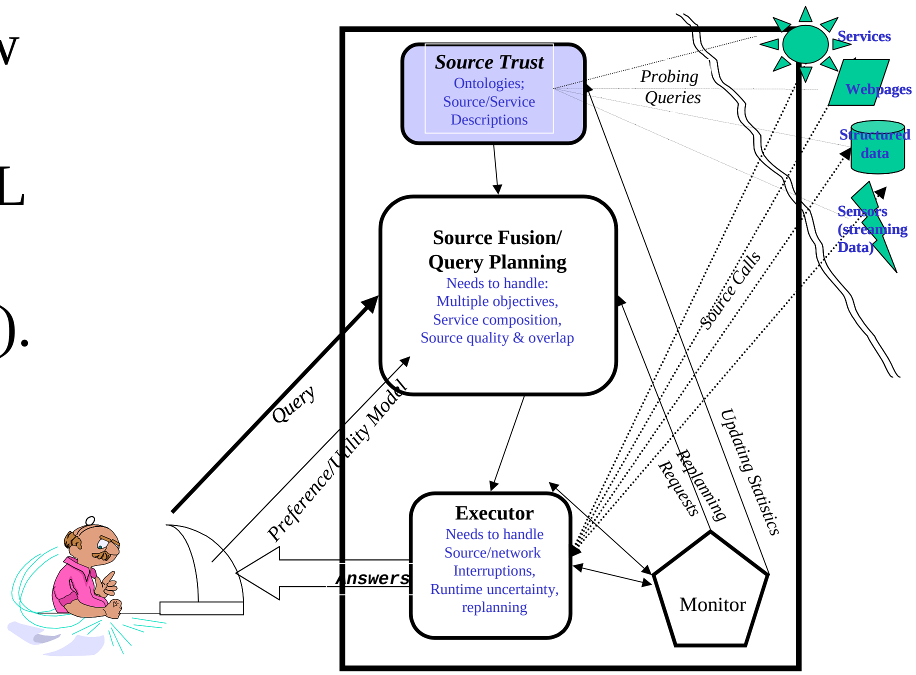


- User queries refer to the *mediated schema*.
- Data is stored in the sources in a *local schema*.
- Content descriptions provide the semantic mappings between the different schemas.
- Mediator uses the descriptions to translate user queries into queries on the sources.



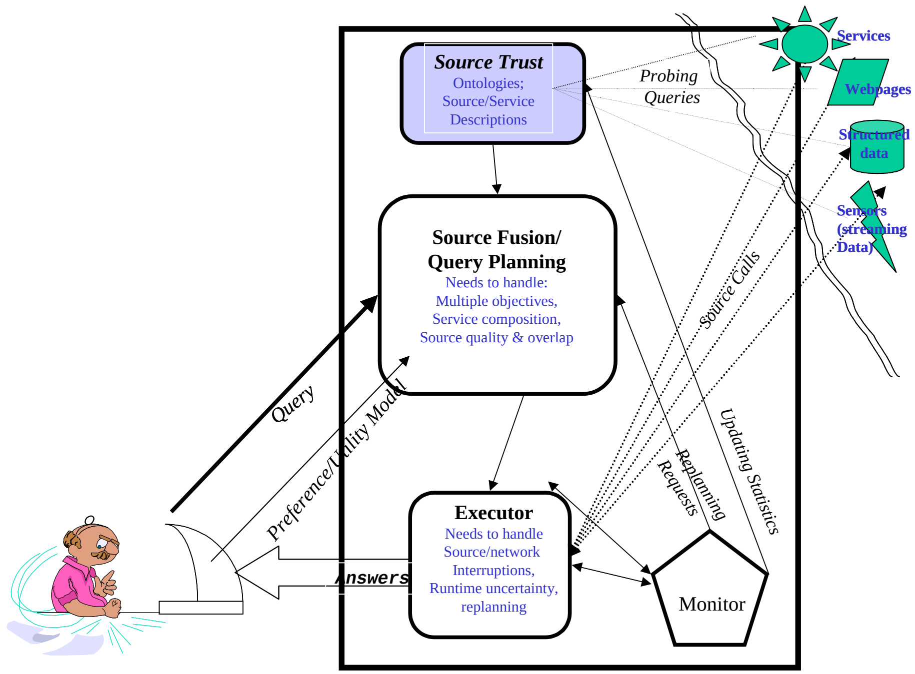
Source Descriptions

- Contains all meta-information about the sources:
  - Logical source contents (books, new cars).
  - Source capabilities (can answer SQL queries)
  - Source completeness (has *all* books).
  - Physical properties of source and network.
  - Statistics about the data (like in an RDBMS)
  - Source reliability
  - Mirror sources
  - Update frequency.
    - Learn this meta-information (or take as input). [Craig]



Source Access

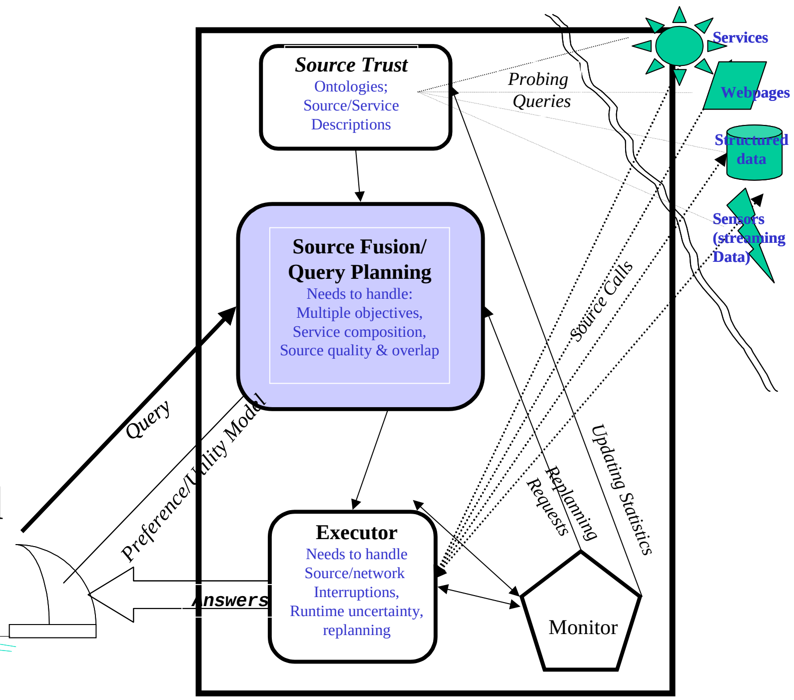
- How do we get the "tuples"?
  - Many sources give "unstructured" output
    - Some inherently unstructured; while others "englishify" their database-style output
  - Need to (un)Wrap the output from the sources to get tuples
  - "Wrapper building"/Information Extraction
  - Can be done manually/semi-manually
    - Craig will talk about this



Source/Data Alignment

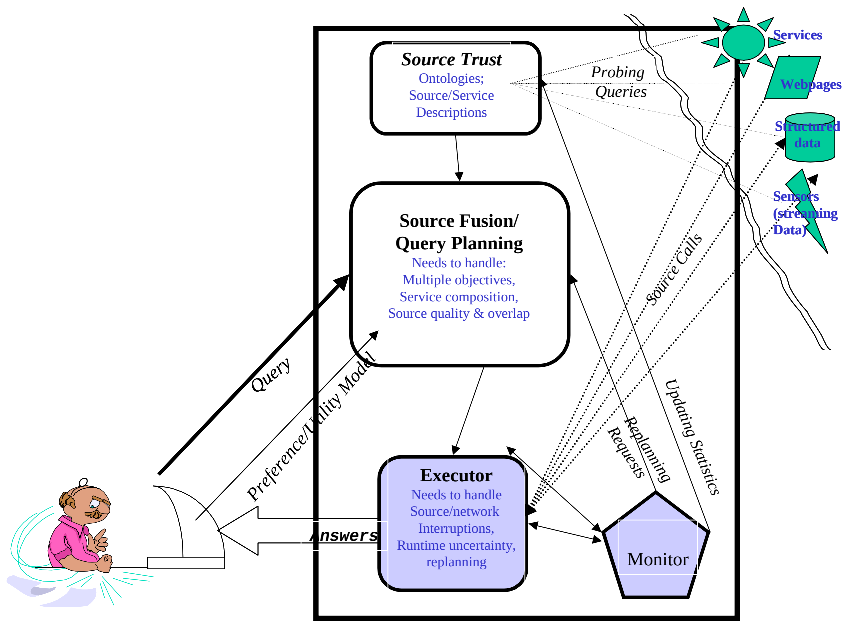
- Source descriptions need to be aligned
  - Schema Mapping problem
- Extracted data needs to be aligned
  - Record Linkage problem

- Two solutions:
  - **Semantic Web solution:** Let the source creators help in mapping and linkage
    - Each source not only exports its schema and gives enough information as to how the schema is related to other "broker" schemas
    - During integration, the mediator chains these relations to align the schemas
  - **Machine Learning solution:** Let the mediator compute the alignment automatically [Craig]



Query Procesing

- Generating answers…
  – Need to reformulate queries onto sources as needed
  – Need to handle imprecision of user queries and incompleteness of data sources.



- Optimizing query processing
  – Needs to handle source overlap, tuple quality, source latency
  – Needs to handle source access limitations

Information Integration & other buzzwords

- XML
  - Can facilitate structured sources delineating their output records syntactically (reducing need for information extraction/screen scraping)
- Semantic Web
  - Can facilitate cooperative sources exposing & mapping their schema information
- Distributed/Multi-databases
  - ..expect much more control over the data sources being integrated
- Data warehouses
  - One way of combining information from multiple sources is to retrieve and store their contents in a single database
- Collection selection
  - ..does “web search” over multiple text collections (and sends pointers rather than answers)
- Mashups
  - ..can be seen as very task-specific information-integration solutions

Information Integration on the Web

AAAI Tutorial
July 22, 2007

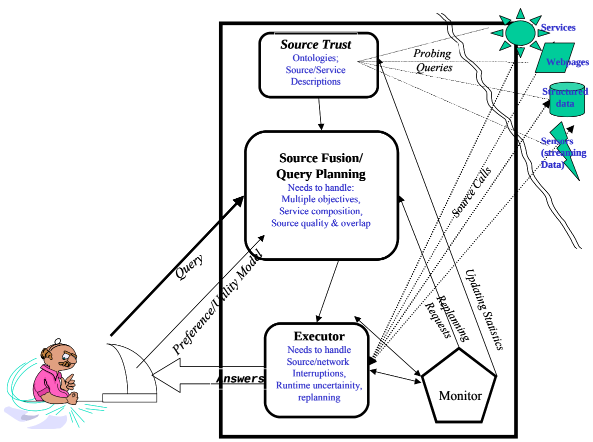
Rao Kambhampati, ASU
Craig Knoblock, USC

Acknowledgements

- Thanks to the many people that provided slides on their work:
  - Ion Muslea
  - Kristina Lerman
  - Andrew McCallum
  - Matthew Michelson
  - AnHai Doan
  - Mark Carman
  - Mikhail Bilenko

Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Wrapper Learning Task

NAME	Casablanca Restaurant
STREET	220 Lincoln Boulevard
CITY	Venice
PHONE	(310) 392-5751

Approaches to Wrapper Construction

- Wrapper learning methods
  - Rule learning [Kushmerick, AIJ 2000]
  - Multi-view learning [Muslea, Minton, & Knoblock, JAIR 2006]
- Methods for automatic wrapper creation and data extraction
  - Grammar Induction approach [Crescenzi & Mecca, JACM 2004]
  - Website structure-based approach
    - *AutoFeed: An Unsupervised Learning System for Generating Webfeeds [Gazen & Minton, AAAI 2006]*
    - *Using the Structure of Web Sites for Automatic Segmentation of Tables [Lerman et al., Sigmod, 2004]*
  - DOM-based:
    - Simile <simile.mit.edu>
    - Dapper <www.dapper.net>

Learning LR extraction rules

<html> Name: Kim's Phone: (800) 757-1111 …

<html> Name: Joe's Phone: (888) 111-1111 …

Learning LR extraction rules

<html> Name: Kim's Phone: (800) 757-1111 …

<html> Name: Joe's Phone: (888) 111-1111 …

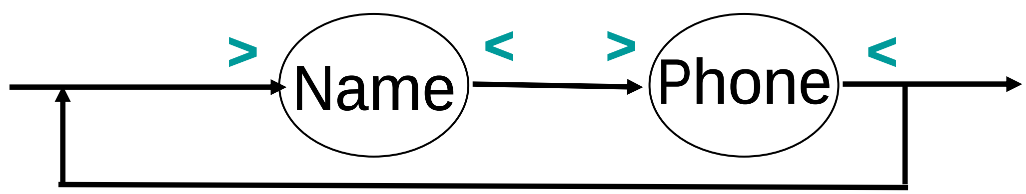
- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct

Learning LR extraction rules

<html> Name: Kim’s Phone: (800) 757-1111 …

<html> Name: Joe’s Phone: (888) 111-1111 …

- Admissible rules:
    - prefixes & suffixes of items of interest
- Search strategy:
    - start with shortest prefix & suffix, and expand until correct

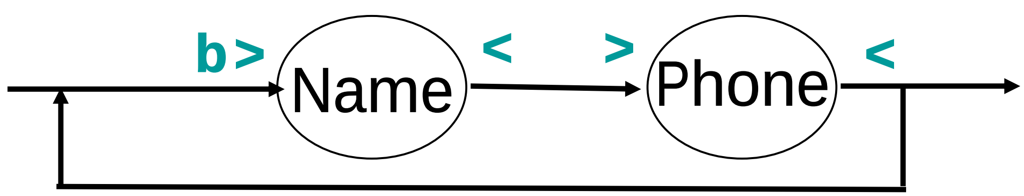


Learning LR extraction rules

<html> Name: Kim’s Phone: (800) 757-1111 …

<html> Name: Joe’s Phone: (888) 111-1111 …

- Admissible rules:
    - prefixes & suffixes of items of interest
- Search strategy:
    - start with shortest prefix & suffix, and expand until correct

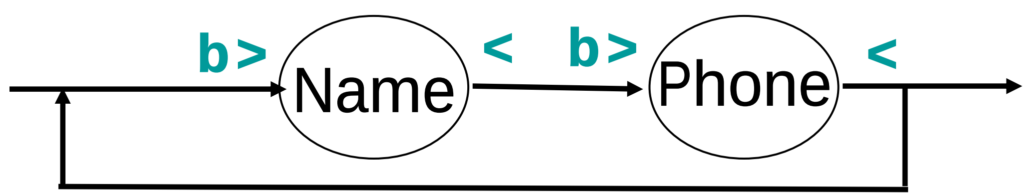


Learning LR extraction rules

<html> Name: Kim's Phone: (800) 757-1111 …

<html> Name: Joe's Phone: (888) 111-1111 …

- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct

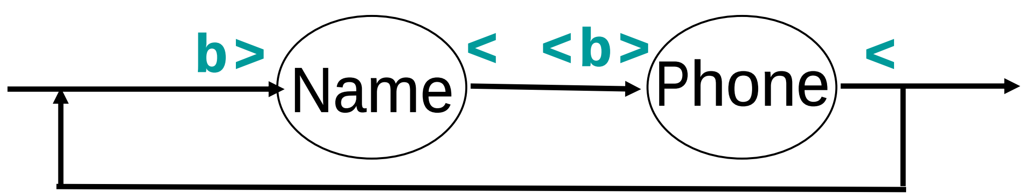


Learning LR extraction rules

<html> Name: Kim's Phone: (800) 757-1111 …

<html> Name: Joe's Phone: (888) 111-1111 …

- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct



Multi-view Learning

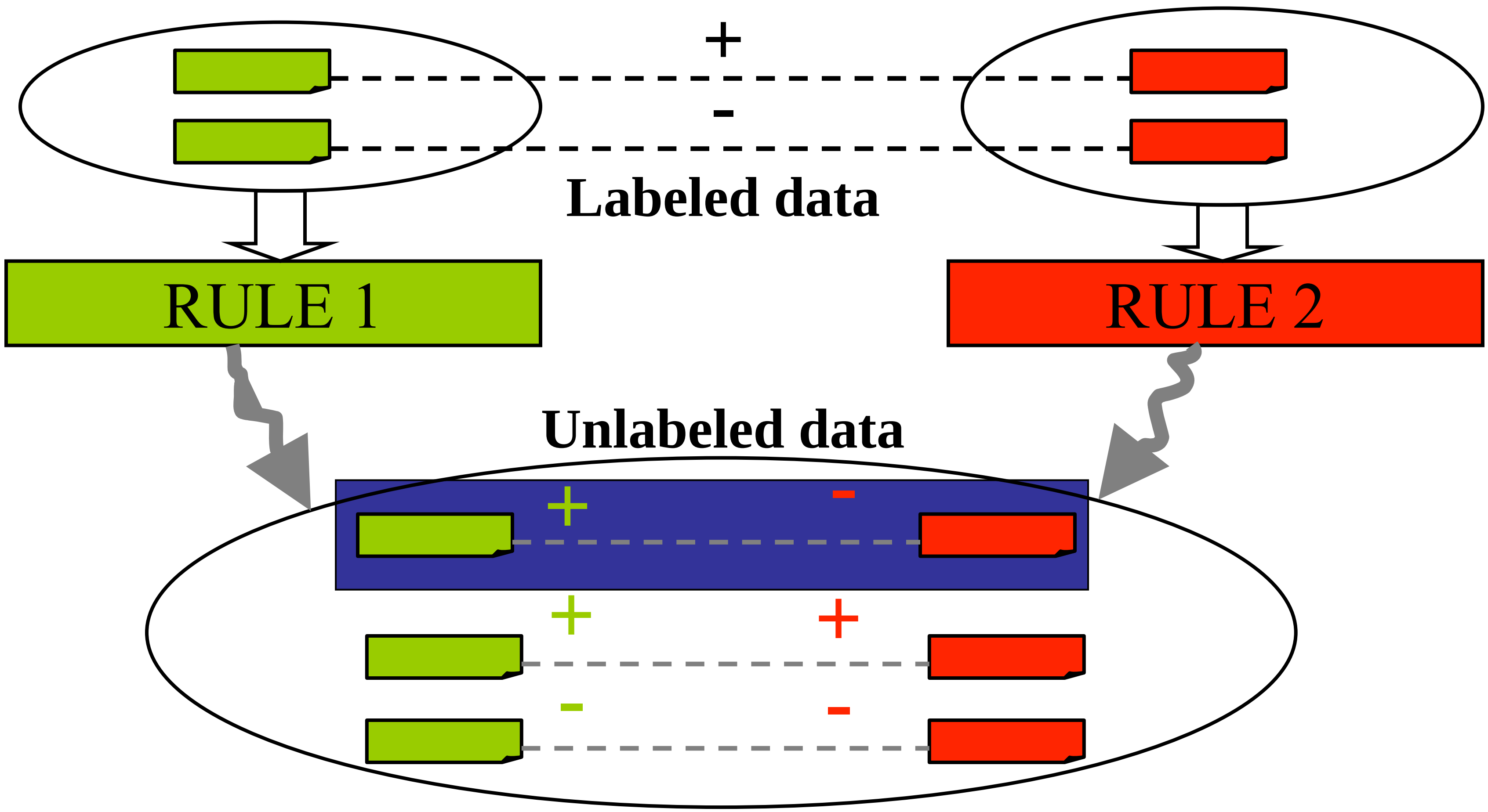
Two ways to find start of the phone number:

SkipTo(**Phone:**)

BackTo(**(*Number*)**)

Name: KFC <p> Phone: (310) 111-1111 <p> Review: Fried chicken …

Multi-view Learning



Multi-view Learning for Wrapper Induction

SkipTo(**Phone:**) BackTo(***(Number)***)

Name: Joel's <p> Phone: (310) 777-1111 <p>Review: ...

Name: Kim's <p> Phone: (213) 757-1111 <p>Review: ...

Name: Chez Jean <p> Phone: (310) 666-1111 <p> Review: ...

Name: Burger King <p> Phone: (818) 789-1211 <p> Review: ...

Name: Café del Rey <p> Phone: (310) 111-1111 <p> Review: ...

Name: KFC <p> Phone: (800) 111-7171 <p> Review:...

Discussion

- Basic problem is to learn how to extract the data from a page
- Range of techniques that vary in the
  - Learning approach
  - Rules that can be learned
  - Efficiency of the learning
  - Number of examples required to learn
- Regardless, all approaches
  - Require labeled examples
  - Are sensitive to changes to sources

Grammar Induction Approach
[Crescenzi, Mecca, & Merialdo]

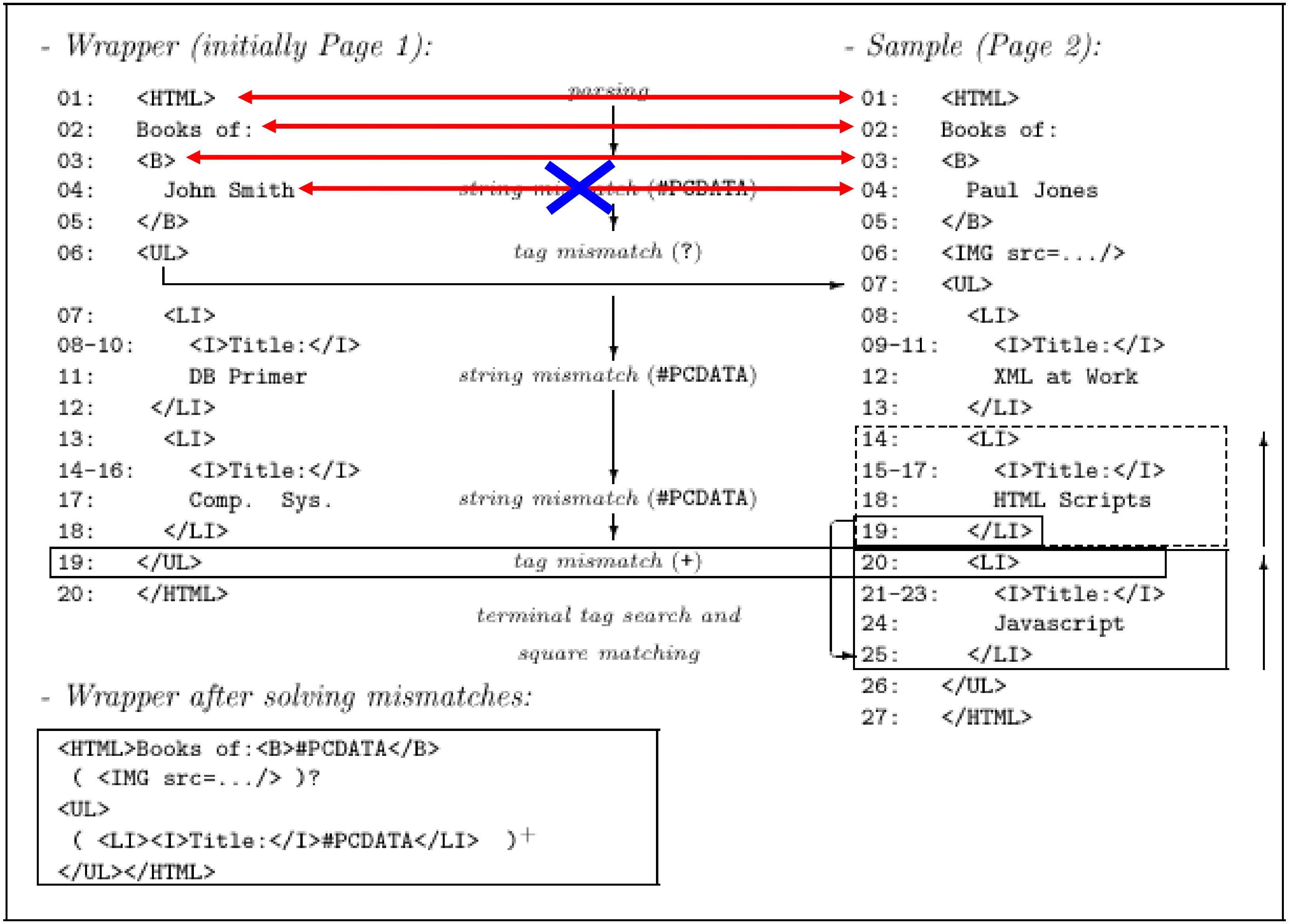
- Given a set of example pages
- Generates a *Union-free Regular Expression (UFRE)*
  - RE without any disjunctions
    - List of tuples (possibly nested): (a, b, c)+
    - Optional attributes: (a)?
  - Strong assumption that usually holds
- Find the least upper bounds on the RE lattice to generate a wrapper in *linear time*
- Reduces to finding the least upper bound on two UFREs

Matching/Mismatches

Given a set of pages of the same type

- Take the first page to be the wrapper (UFRE)
- Match each successive sample page against the wrapper
- Mismatches result in generalizations of the regular expression
- Types of mismatches:
  - String mismatches
  - Tag mismatches

Example Matching



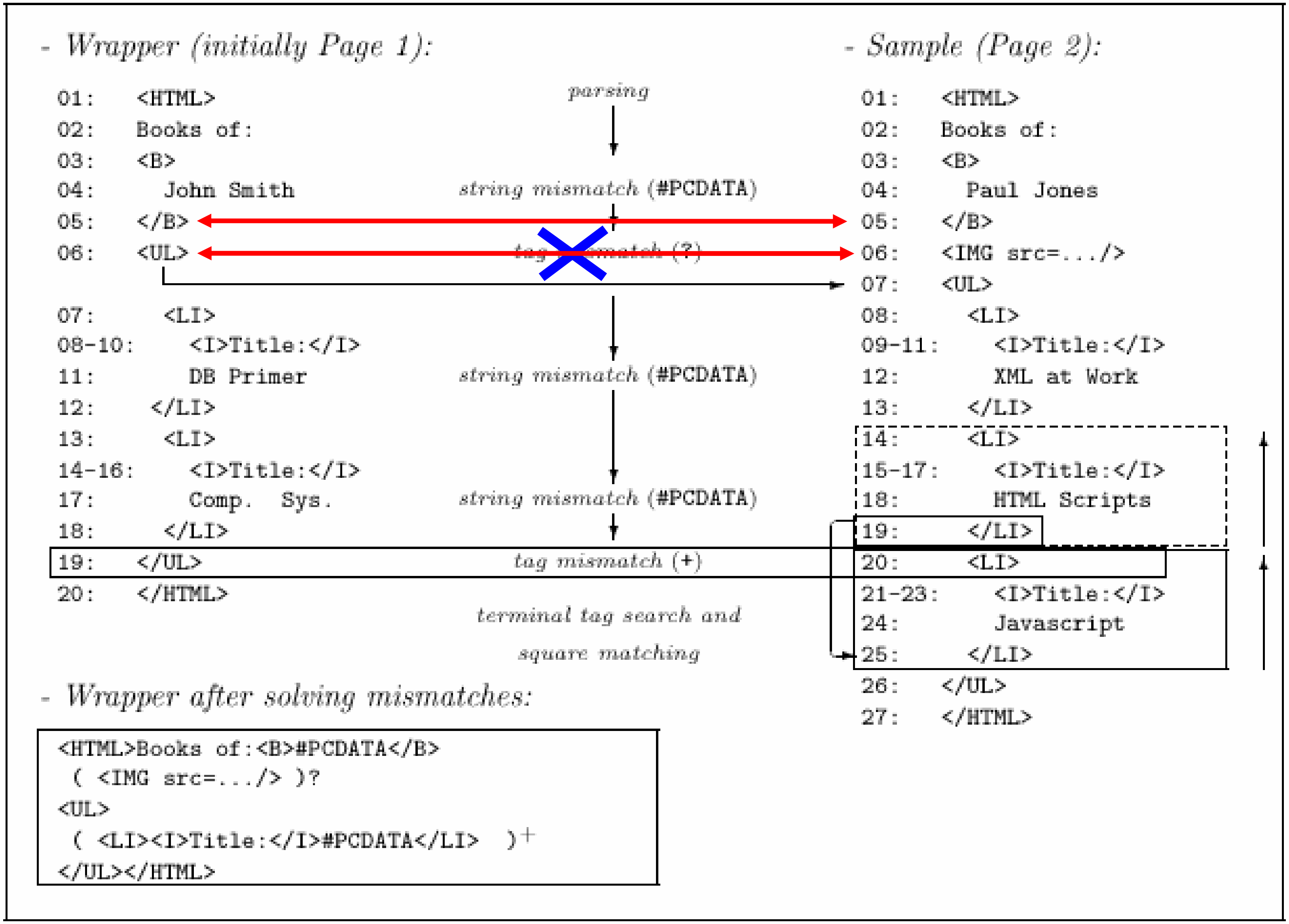
String Mismatches: Discovering Fields

- String mismatches are used to discover fields of the document
- Wrapper is generalized by replacing “John Smith” with #PCDATA

<HTML>Books of: <B>John Smith

→ <HTML> Books of: <B>#PCDATA

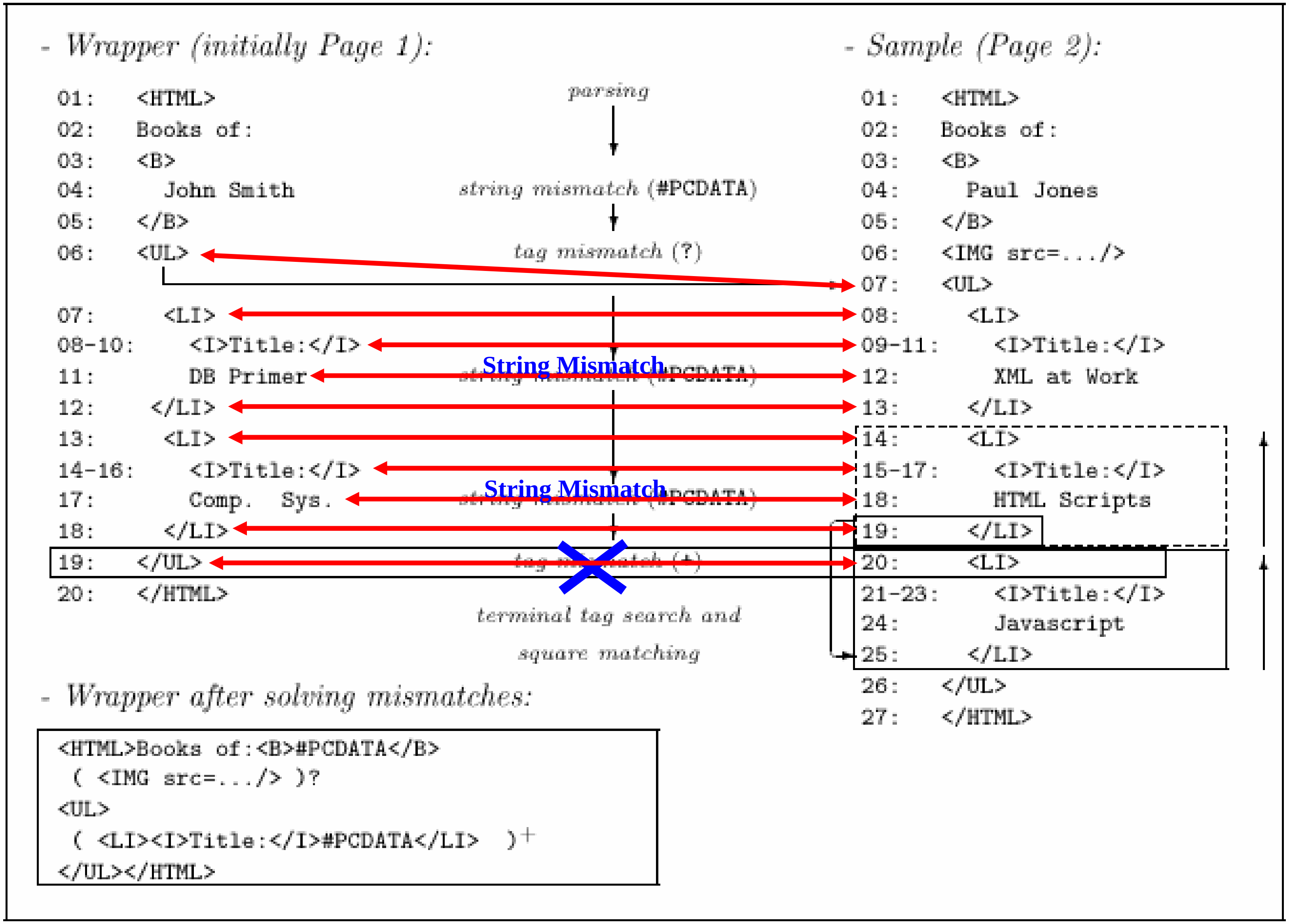
Example Matching



Tag Mismatches: Discovering Optionals

- First check to see if mismatch is caused by an iterator (described next)
- If not, could be an optional field in wrapper *or* sample
- Cross search used to determine possible optionals
- Image field determined to be optional:
  - ( )?

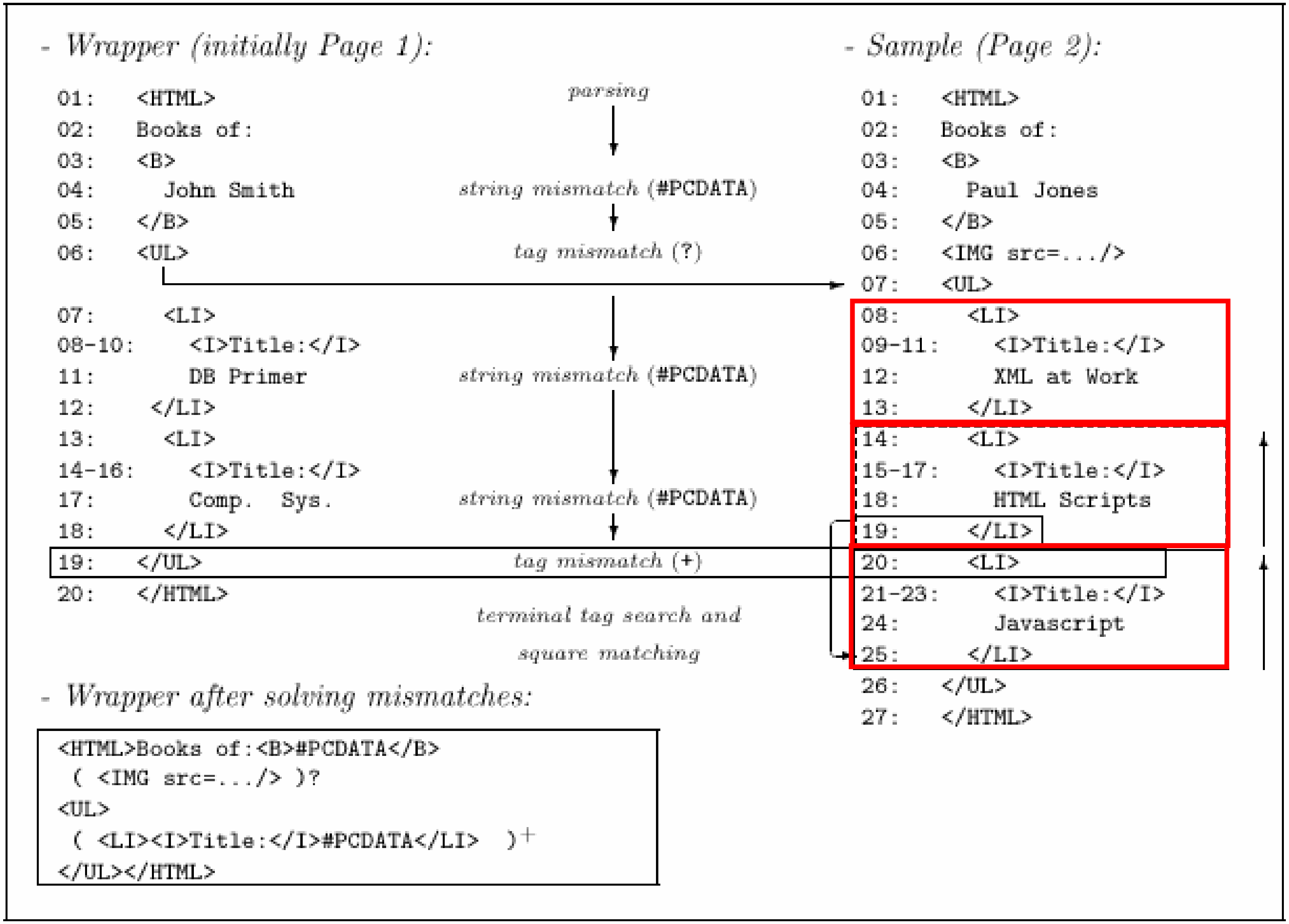
Example Matching



Tag Mismatches: Discovering Iterators

- Assume mismatch is caused by repeated elements in a list
  - End of the list corresponds to last matching token: </LI>
  - Beginning of list corresponds to one of the mismatched tokens: <LI> or </UL>
  - These create possible “squares”
- Match possible squares against earlier squares
- Generalize the wrapper by finding all contiguous repeated occurences:
  - (<LI><I>Title:</I>#PCDATA</LI>)+

Example Matching

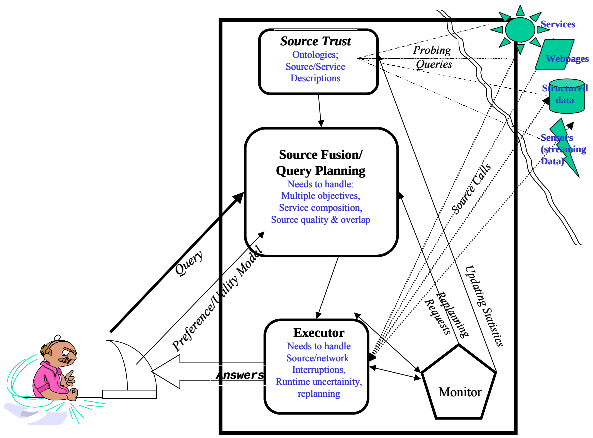


Discussion

- Learnable grammars
  - Union-Free Regular Expressions (RoadRunner)
    - Variety of schema structure: tuples (with optional attributes) and lists of (nested) tuples
    - Does not efficiently handle disjunctions – pages with alternate presentations of the same attribute
- Assumptions:
  - Pages are well-structured
  - Want to extract at the level of entire fields
  - Structure can be modeled without disjunctions

Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



What is “Information Extraction”

As a task: **Filling slots in a database from sub-segments of text.**

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

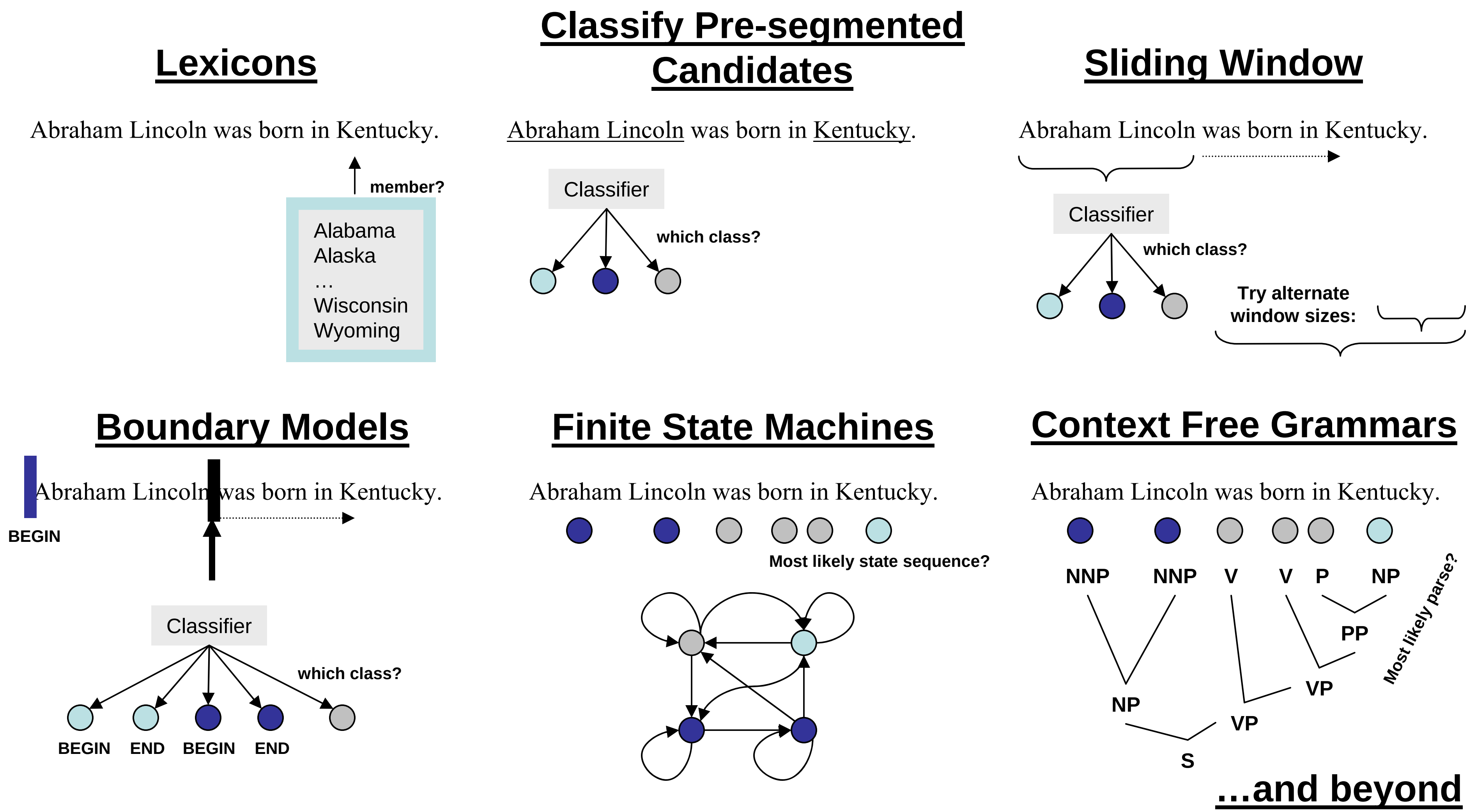
"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…

IE

NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Landscape of IE Techniques



Any of these models can be used to capture words, formatting or both.

Extraction from Ungrammatical & Unstructured Text

[Michelson & Knoblock, IJCAI '05]

Page 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

	Topic	Replies	Last Comment	Started B
	SACRAMENTO HOTEL LIST	0	11/21/04 9:56 pm	westcoastma
	3* Rancho Cordova Holiday Inn $35, 1 nite (12/11)	1	12/9/04 12:37 am	future canadi
	3* Doubletree Sacto Arden 12/11 1 Night $34	1	12/7/04 4:46 pm	OCTraveler
	4* Sacramento Failed Bid $85 12/7	1	12/6/04 6:29 pm	Sheryl
	Failed bid Sacramento Downtown 12/6 for 1 night, 4*	13	12/6/04 6:25 pm	emaij
	2.5* Wingate Inn Rancho Cordova 5/10-5/13/05 $32	0	12/4/04 7:11 pm	ego68
	3* DoubleTree Sacramento $35 (12/04/04)	0	11/30/04 11:34 pm	shizzolator
	2.5* Rancho Cordova Wingate Inn $32 (11/23-25)	1	11/27/04 12:19 pm	Profiler
	4* DT Hyatt 11/21 $60 11/23 $60; Sheraton Grand 11/25 $55	0	11/22/04 1:22 pm	bonish
	3* Doubletree Arden/Sacramento $37 11/19	1	11/20/04 1:53 am	ahallez
	2.5* Wingate Inn Rancho Cordova $33 11/13	2	11/19/04 1:44 am	cykick42
	2.5* DT Hawthorne Suites $40 (11/18-20)	0	11/18/04 10:08 pm	Colfax30
	Roseville 2.5*Larkspur $72(11/22-24) 2* Fairfield $80(11/24)	2	11/17/04 4:38 pm	mcrinca
	3* Rancho Cordova Holiday Inn $32 (11/17)	0	11/16/04 10:20 pm	Colfax30
	3* Doubletree Sacramento $40 (11/11)	2	11/16/04 11:05 am	OCTraveler
	3* Doubletree Sacramento Arden $36 11/24	0	11/15/04 1:04 am	bomawin

Ungrammatical & Unstructured Text

For simplicity → “posts”

Goal:

<hotelArea>univ. ctr.</hotelArea>

Beware 2* at the airport!!!!	2	7/18/00 1:25 am
$25 winning bid at holiday inn sel. univ. ctr.	1	6/26/00 1:48 pm
3* Holiday Inn North-McKnight Rd, $10+20, 1/19	3	1/27/01 6:34 pm

<price>$25</price> <hotelName>holiday inn sel.</hotelName>

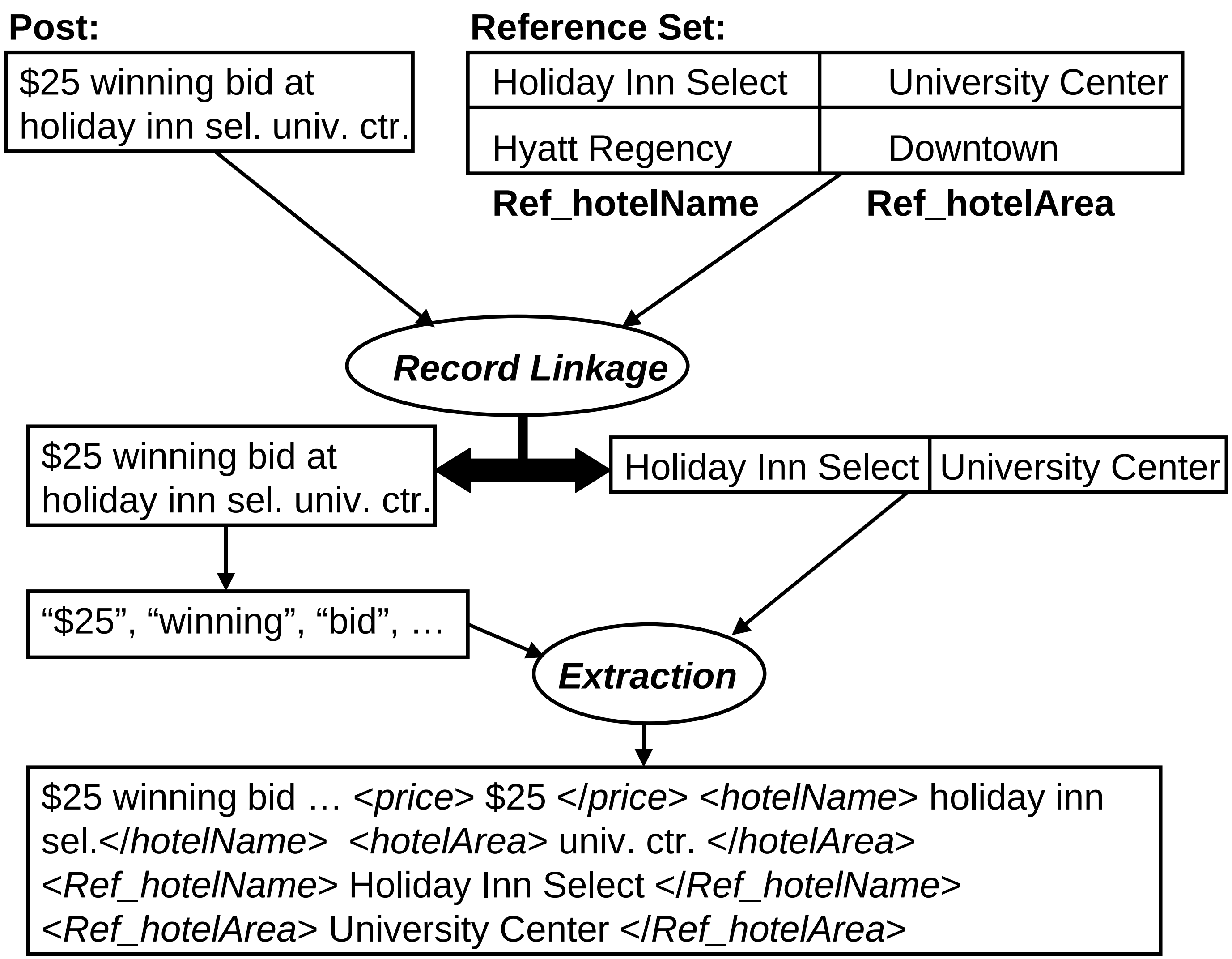
Reference Sets

IE infused with outside knowledge (lexicon)

"Reference Sets"

- Collections of known entities and the associated attributes
- Online (offline) set of docs
  - CIA World Fact Book
- Online (offline) database
  - Comics Price Guide, Edmunds, etc.
- Build from ontologies on Semantic Web

Algorithm Overview – Use of Ref Sets



Record Linkage Problem

- ***Posts not yet decomposed attributes.***
- ***Extra tokens that match nothing in Ref Set.***

Post:

“$25 winning bid at $\underbrace{\text{holiday inn sel.}}_{\textit{hotel name}}$ $\underbrace{\text{univ. ctr.}}_{\textit{hotel area}}$”

Reference Set:

Holiday Inn	Greentree
Holiday Inn Select	University Center
Hyatt Regency	Downtown
hotel name	*hotel area*

Record Linkage Approach

***P* = "$25 winning bid at holiday inn sel. univ. ctr."**

Record Level Similarity + Field Level Similarities

V_{RL} = < ***RL_scores***(*P*, "Hyatt Regency Downtown"),

RL_scores(*P*, "Hyatt Regency"),

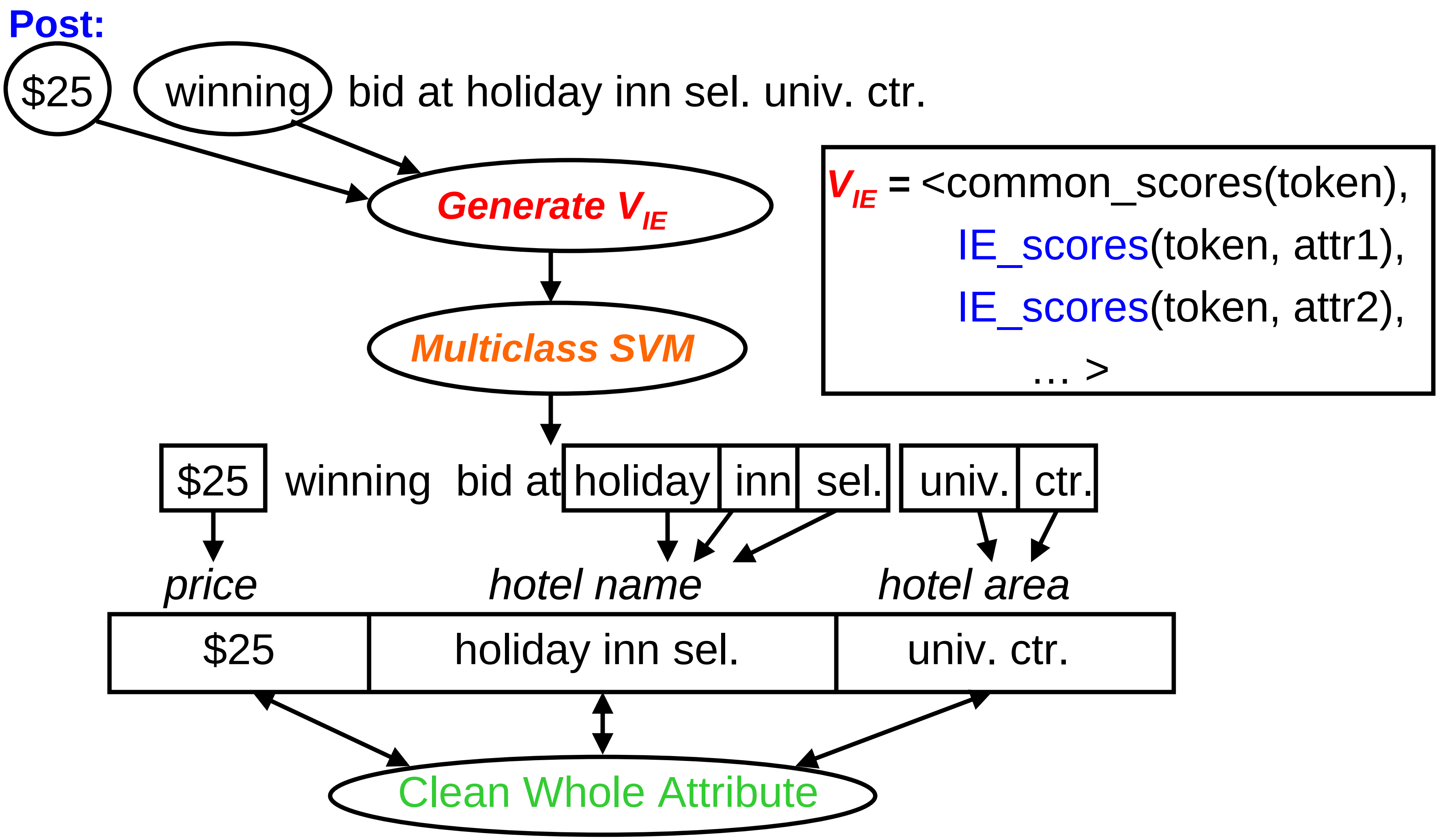
RL_scores(*P*, "Downtown")>

Binary Rescoring

SVM

Best matching member of the reference set for the post

Extraction Algorithm

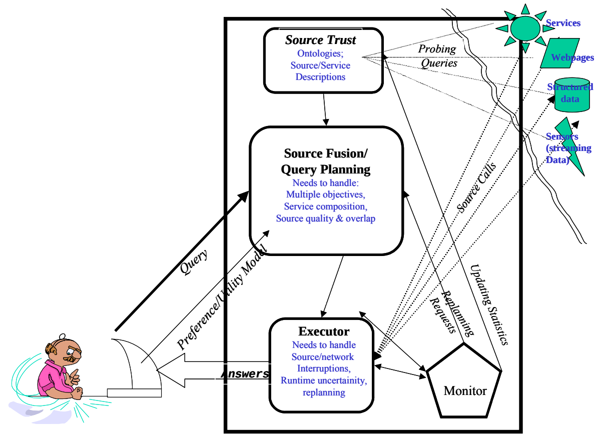


Extraction results: Summary

	Token Level			*Hotel*	Field Level		
	Prec.	Recall	F-Mes.		Prec.	Recall	F-Mes.
Phoebus	93.60	91.79	**92.68**		87.44	85.59	**86.51**
Simple Tagger	86.49	89.13	87.79		79.19	77.23	78.20
Amilcare	86.12	86.14	86.11		85.04	78.94	81.88
	Token Level			*Comic*	Field Level		
	Prec.	Recall	F-Mes.		Prec.	Recall	F-Mes.
Phoebus	93.24	84.48	**88.64**		81.73	80.84	**81.28**
Simple Tagger	84.41	86.04	85.43		78.05	74.02	75.98
Amilcare	87.66	81.22	84.29		90.40	72.56	80.50

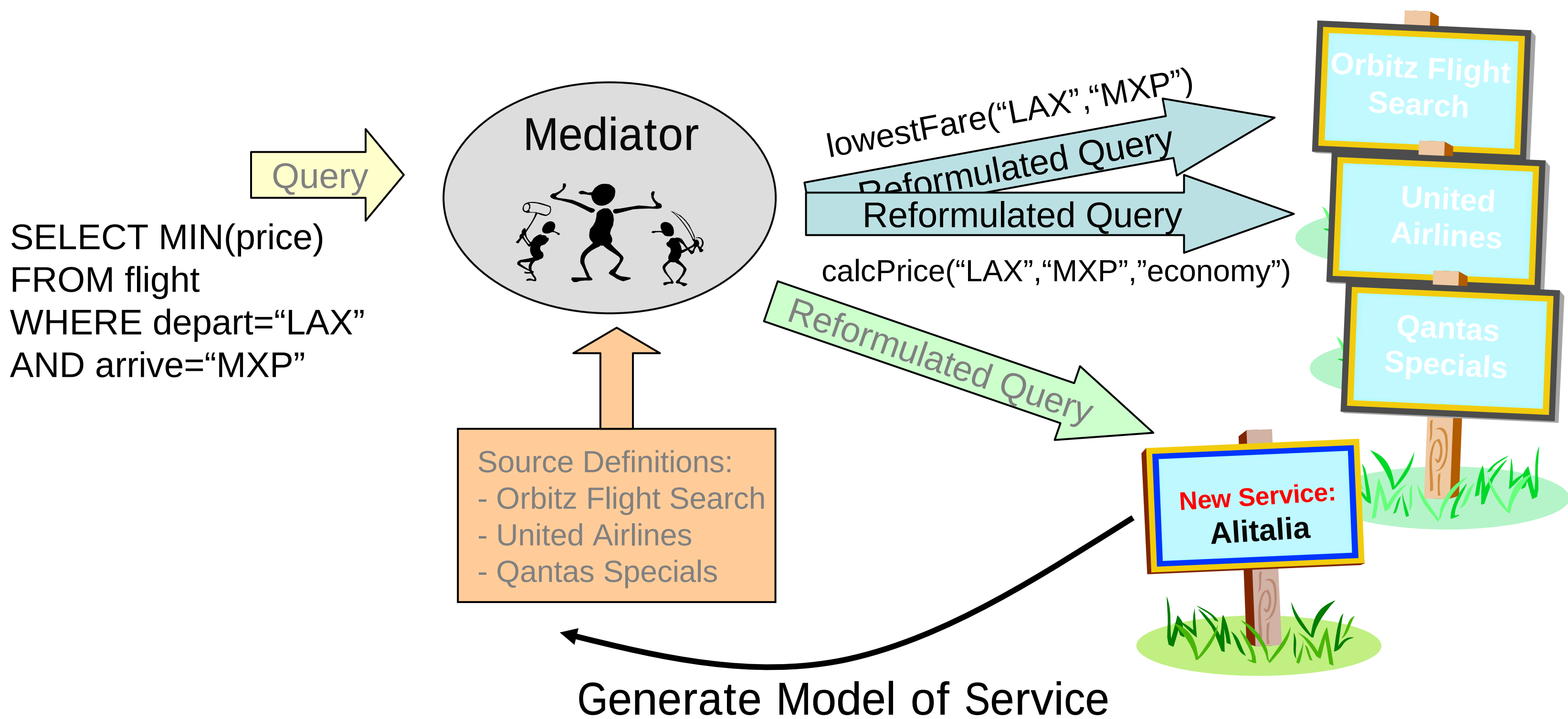
Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Data Integration Systems Require Source Definitions

- New service => no definition!
- Can we model it automatically?



Schema Matching vs. Source Modeling

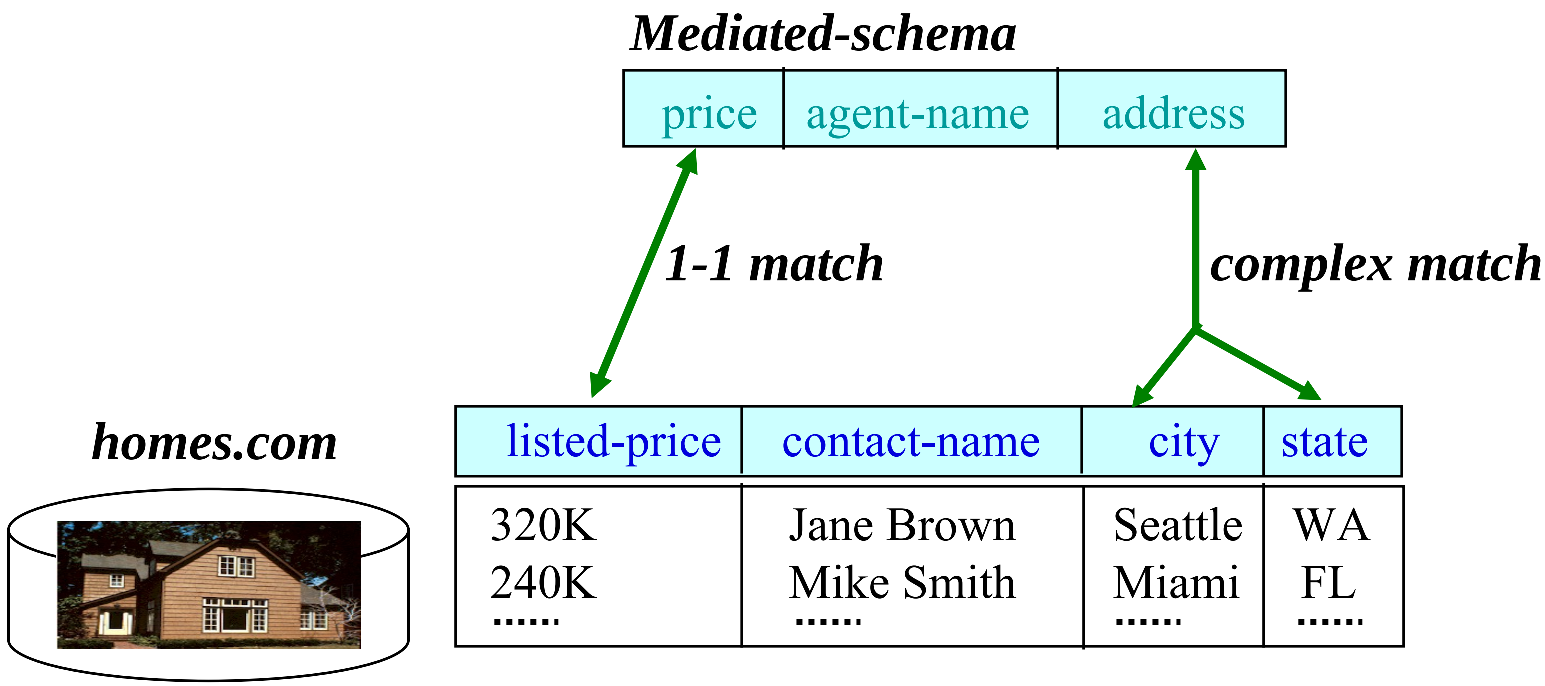
- Schema matching is used when you have a set of databases and need to map them into a unified schema
  - Model of the individual attributes

- Source modeling is used when you have a web service or web form and you want to model the function performed by the service
  - Made possible using the mapping from inputs to outputs
  - Problem is harder because the data is not directly available, but must be queried

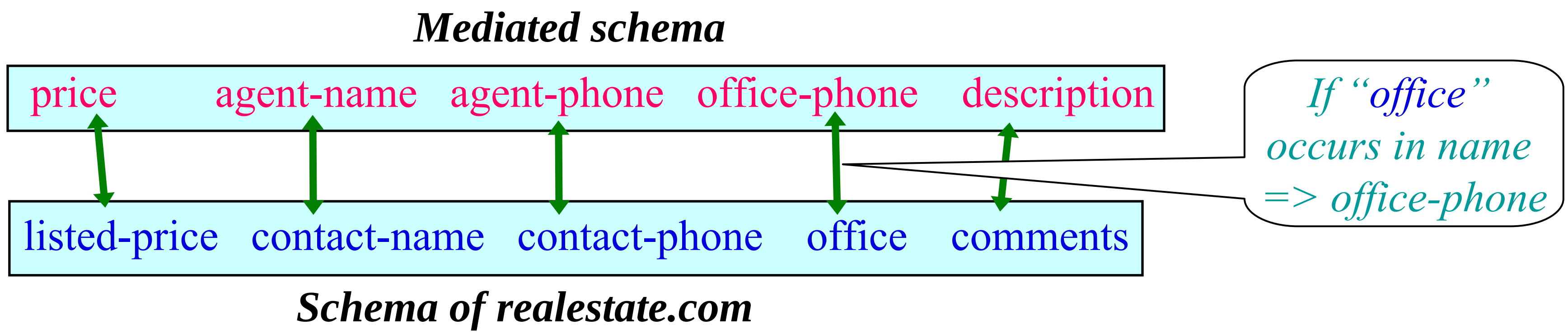
Approaches to Schema Matching

- There is 20 years of research on schema matching in DB
  - Most of it ignores the data!
  - Here is an excellent survey:
    - A Survey of Approaches to Automatic Schema Matching (2001), Erhard Rahm, Philip A. Bernstein, VLDB Journal: Very Large Data Bases
- Multi-strategy learning for schema matching
  - LSD System (Doan, Domingos, and Halevy, ML 2003)

Semantic Matches between Schemas



Must Exploit Multiple Types of Information!



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

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

homes.com

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

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

Base Learners

- Training

Object

Observed label

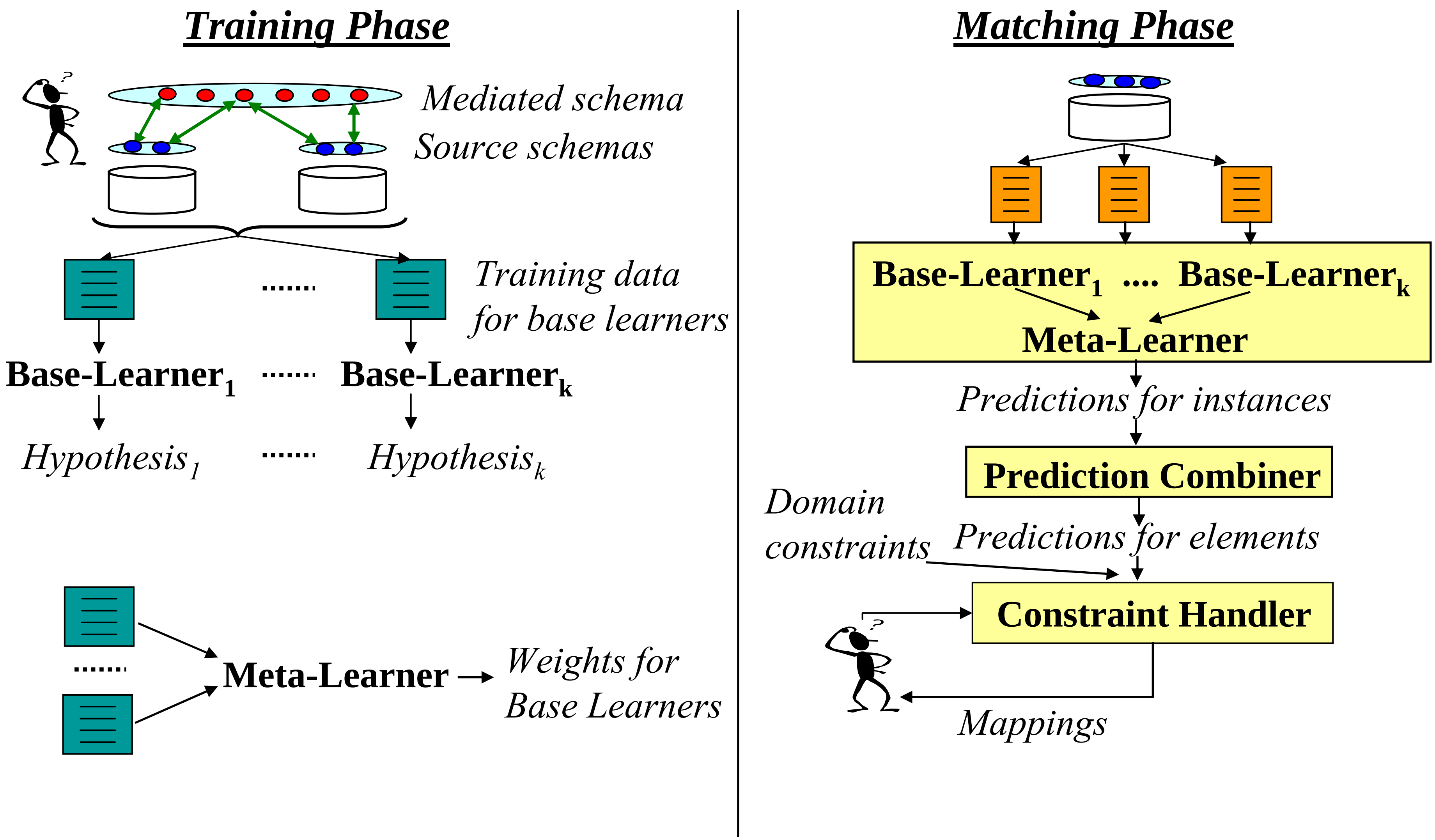
Training examples

(X_1, C_1)
(X_2, C_2)
...
(X_m, C_m)

$\Rightarrow$ Classification model (hypothesis)

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

The LSD Architecture



Training the Base Learners

Mediated schema

address	price	agent-name	agent-phone	office-phone	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
....	....	....	....	....	....

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)
....

Meta-Learner: Stacking
[Wolpert 92,Ting&Witten99]

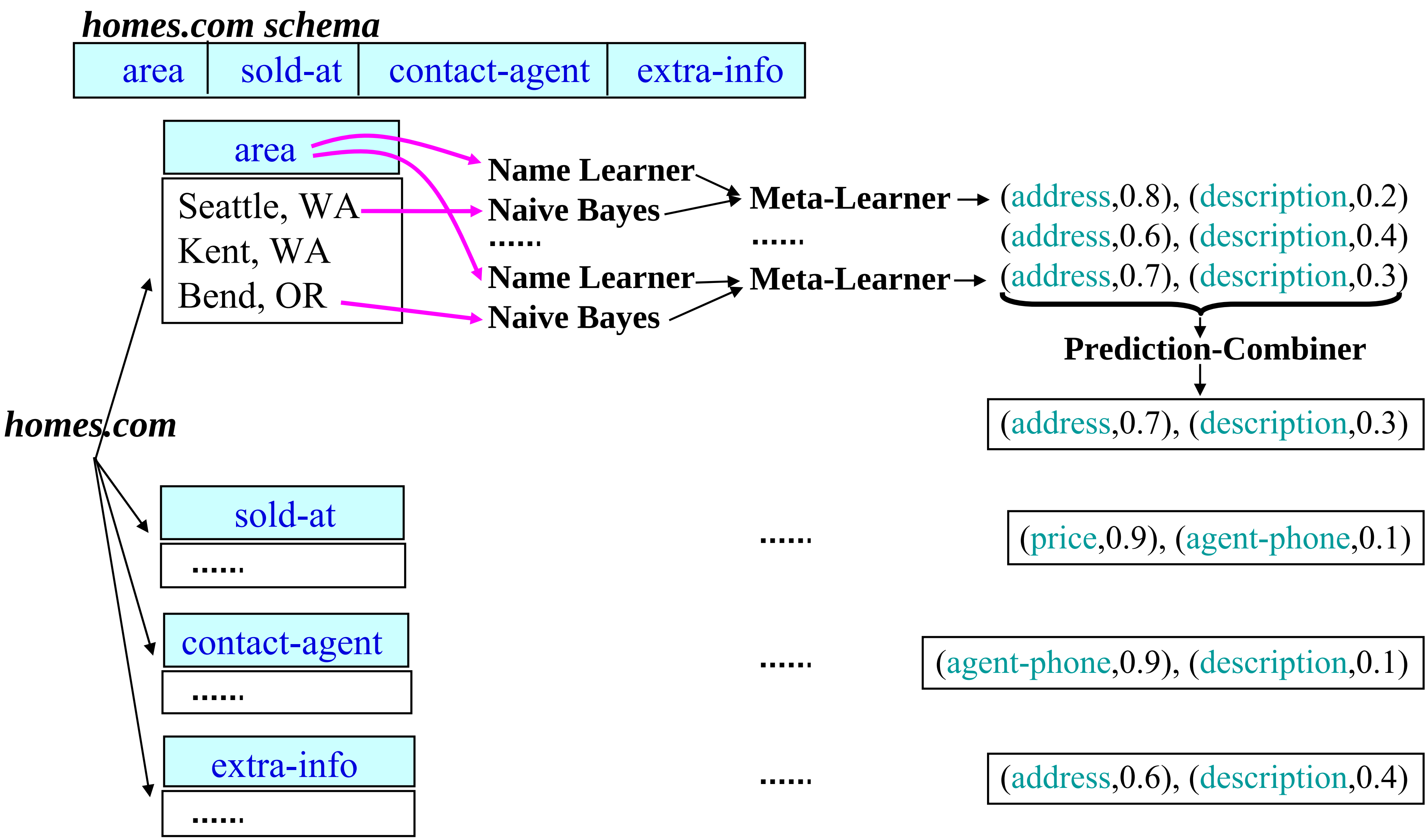
- Training
  - uses training data to learn weights
  - one for each (base-learner,mediated-schema element) pair
  - weight (Name-Learner,address) = 0.2
  - weight (Naive-Bayes,address) = 0.8
- Matching: combine predictions of base learners
  - computes weighted average of base-learner confidence scores

area
Seattle, WA Kent, WA Bend, OR

Name Learner → (address,0.4)
Naive Bayes → (address,0.9)

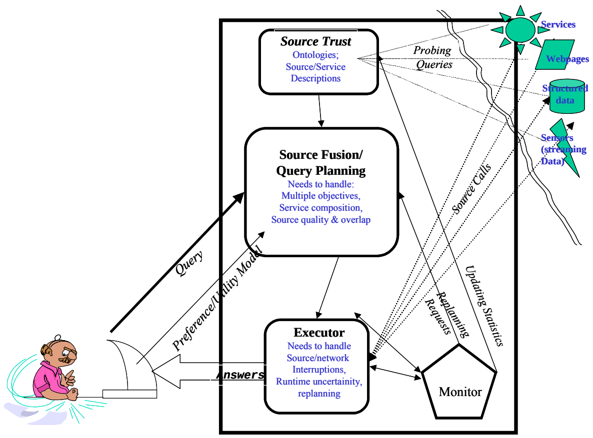
Meta-Learner → (address, 0.4*0.2 + 0.9*0.8 = 0.8)

Applying the Learners



Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Approaches to Modeling Sources

Step 1: Semantic Labeling

Classify input & output *semantic types*

- Metadata-based classification of data types used by Web services (Hess & Kushmerick, ISWC 2003)
- Woogle: Metadata-based clustering of data and operations used by Web services (Dong et al, VLDB 2004)
- Learn semantic types (Lerman et al., AAAI 2006)

Step 2: Functional Modeling

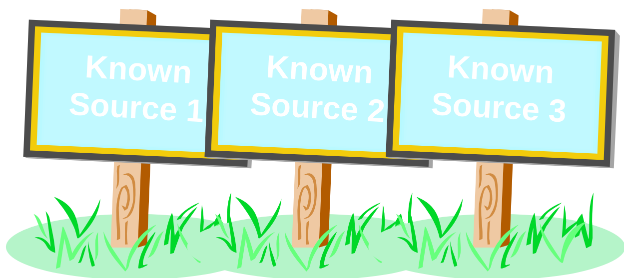
Model the *functionality* of service

- Learn functions describing operations on internet

(Perkowitz & Etzioni, IJCAI 1995)

Learn function of a web service (Carman & Knoblock, IJCAI 2007)

Modeling Sources: an Example



source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

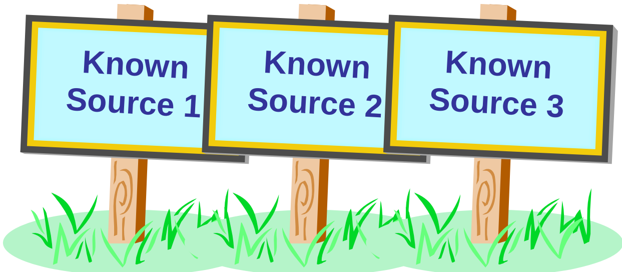
Step 1:

classify input & output semantic types, using:

- Metadata (labels)
- Data (content)

source4($startZip, $endZip, separation)

Modeling Sources: Step 2



source1($zip, lat, long) :-
  centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
  greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
  convertKm2Mi(dist1, dist2).

Step 2:

model functionality of service by:

- generating plausible definitions

source4($zip1, $zip2, dist) :-

source1(zip1, lat1, long1),
source1(zip2, lat2, long2),
source2(lat1, long1, lat2, long2, dist2),
source3(dist2, dist).

centroid(zip1, lat1, long1),
centroid(zip2, lat2, long2),
greatCircleDist(lat1, long1, lat2, long2, dist2),
convertKm2Mi(dist1, dist2).

Modeling Sources: Step 2

Step 2:

model functionality of service by:

- generating plausible definitions
- comparing the output they produce

```
source4( $zip1, $zip2, dist) :-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).
```

match

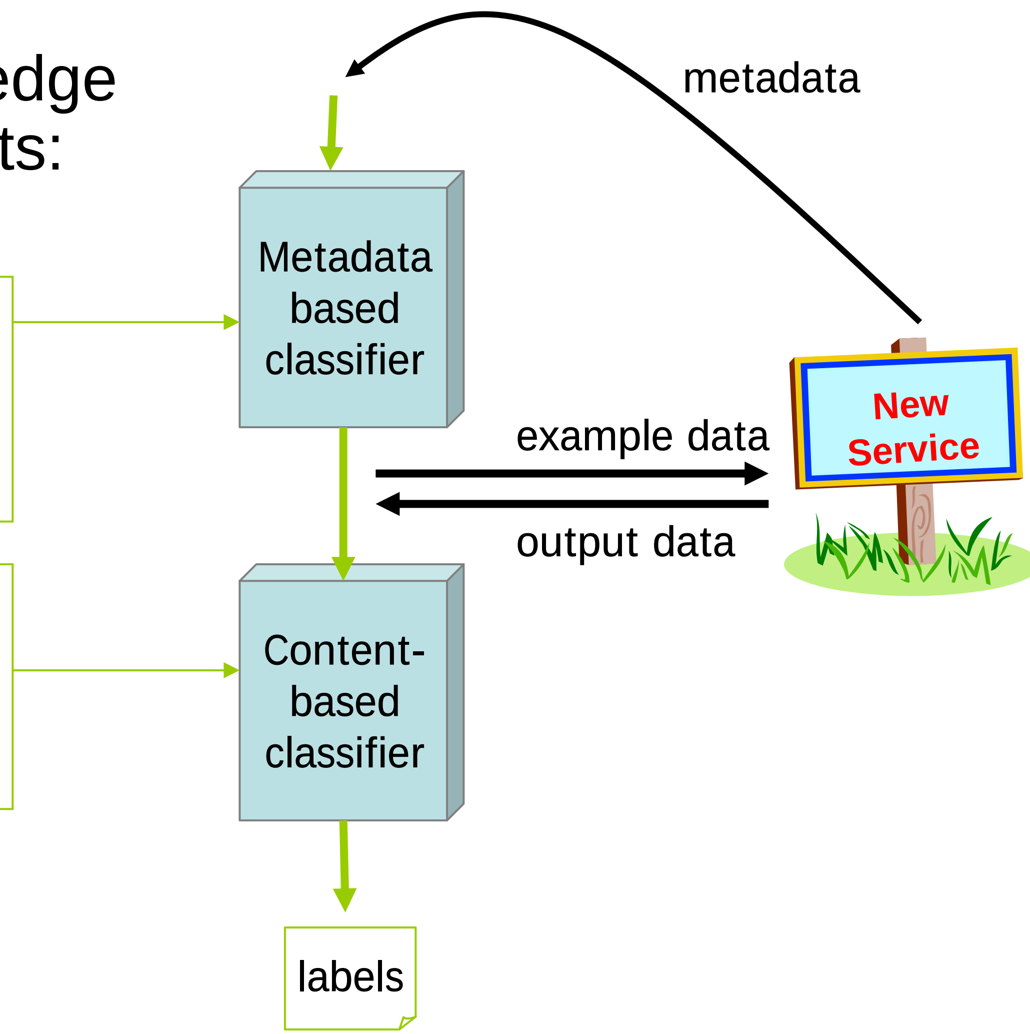
$zip1	**$zip2**	**dist *(actual)***	**dist *(predicted)***
80210	90266	842.37	843.65
60601	15201	410.31	410.83
10005	35555	899.50	899.21

Approach to Semantic Labeling

Leverage existing knowledge
to label inputs & outputs:

<u>Labeled Example WSDL Files:</u>
Operation = “GetZipCodeCoordinates”
    Input = “zip” <Zipcode>
    Output1 = “LatDegrees” <Latitude>
    Output2 = “LonDegrees” <Longitude>

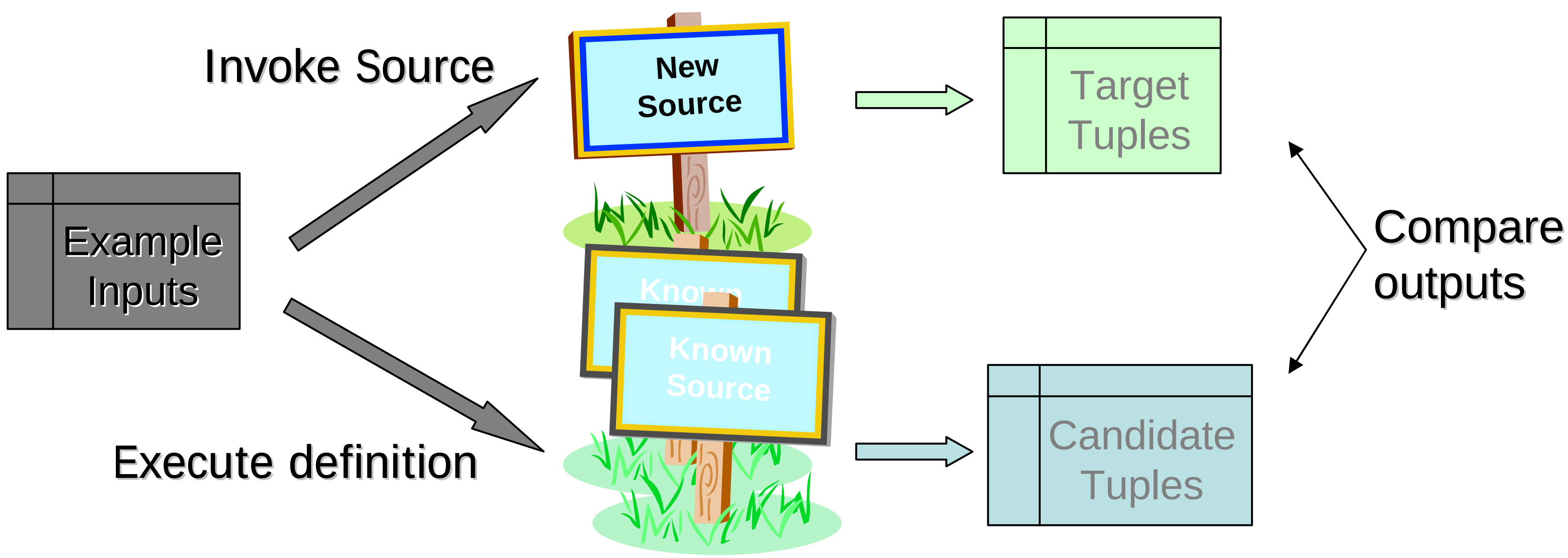
<u>Semantic Types with Examples:</u>
Zipcode: “90066”, ...
Latitude: “34.12”, ...
Temperature: “30°F”, ...
Humidity: “35%”, ...



Functional Modeling

Model the *functionality* of service by:

- Searching through the space of plausible definitions
- Score definitions by comparing the output they produce with that of the source being modeled

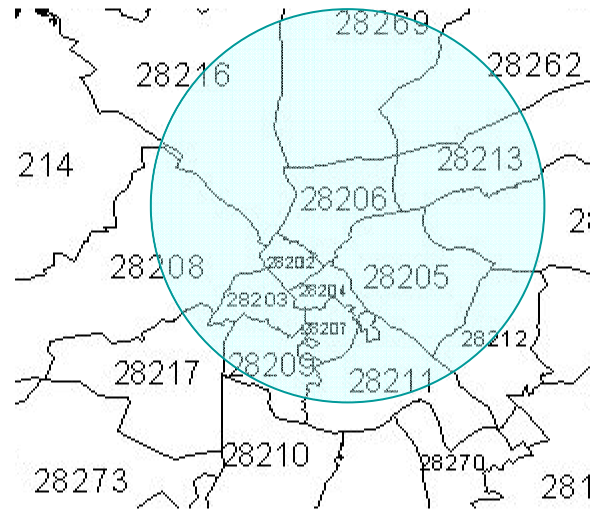


Invoking the Target

Generate Input Tuples: *<zip1, dist1>*

Invoke

source5($zip1, $dist1, zip2, dist2)



Invoke source with *representative* values

- Randomly generate input tuples
  - Use distribution if available
- If no output is produced:
  - Try invoking other sources to generate input

Input <zip1, dist1>	Output <zip2, dist2>
<07307, 50.94>	{<07097, 0.26>, <07030, 0.83>, <07310, 1.09>, ...}
<60632, 10874.2>	{}

randomly generated input tuples

Non-empty Result

Empty Result

Best-first Enumeration of Candidates

Start with empty clause & specialize it by:

- Adding a predicate from set of sources
- Checking that each definition is executable & not redundant

source5($zip1,$dist1,zip2,dist2)

source5(_,_,_,_).

Expand

source5(zip1,_,_,_) :- **source4**(zip1,zip1,_).
source5(zip1,_,zip2,dist2) :- **source4**(zip2,zip1,dist2).
source5(_,dist1,_,dist2) :- **<**(dist2,dist1).
…

Expand

source5(zip1,dist1,zip2,dist2) :- **source4**(zip2,zip1,dist2), **source4**(zip1,zip2,dist1).
source5(zip1,dist1,zip2,dist2) :- **source4**(zip2,zip1,dist2), **<**(dist2,dist1).
…

Evaluating Candidates

- Compare output of each candidate with that of target.
- Average results across different input tuples.

Input **<$zip1, $dist1>**	**Target Output** **<zip2, dist2>**	**Clause Output** **<zip2, dist2>**
<60632, 874.2>	{}	{<60629, 2.15>, <60682, 2.27>, <60623, 2.64>, ..}
<07307, 50.94>	{<07097, 0.26>, <07030, 0.83>, <07310, 1.09>, ...}	{}
<28041, 240.46>	{<28072, 1.74>, <28146, 3.41>, <28138, 3.97>,…}	{<28072, 1.74>, <28146, 3.41>}

No Overlap

No Overlap

Overlap!

Actual Learned Examples

1 **GetDistanceBetweenZipCodes**($zip0, $zip1, dis2):-
GetCentroid(zip0, lat1, lon2), **GetCentroid**(zip1, lat4, lon5),
GetDistance(lat1, lon2, lat4, lon5, dis10), **ConvertKm2Mi**(dis10, dis2).

2 **USGSElevation**($lat0, $lon1, dis2):-
ConvertFt2M(dis2, dis1), **Altitude**(lat0, lon1, dis1).

Distinguished forecast from current conditions

3 **YahooWeather**($zip0, cit1, sta2, , lat4, lon5, day6, dat7,tem8, tem9, sky10) :-
WeatherForecast(cit1,sta2,,lat4,lon5,,day6,dat7,tem9,tem8,,,sky10,,,),
GetCityState(zip0, cit1, sta2).

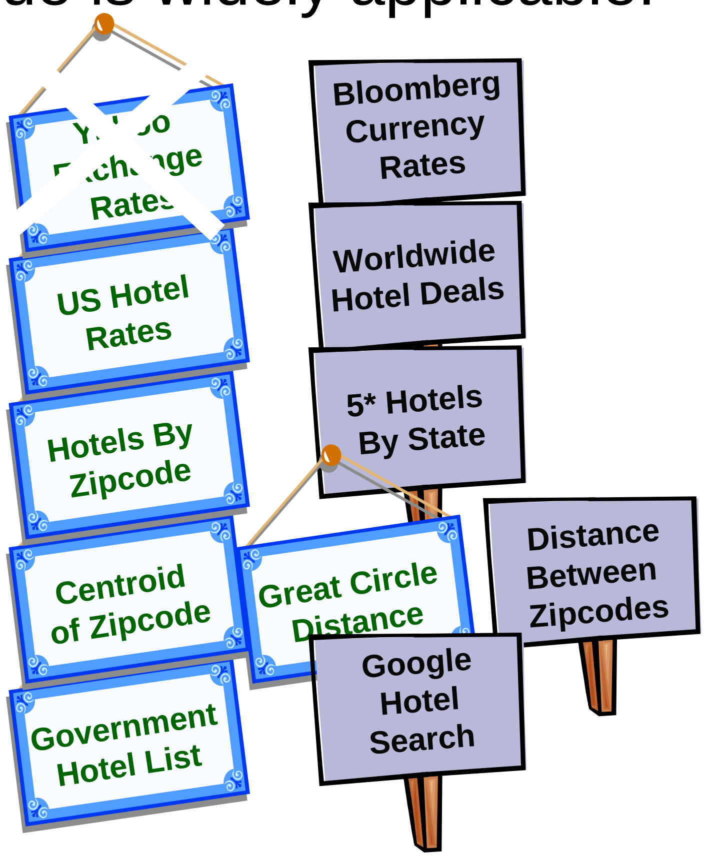
current price = yesterday's close + change

4 **GetQuote**($tic0,pri1,dat2,tim3,pri4,pri5,pri6,pri7,cou8,,pri10,,,pri13,,com15) :-
YahooFinance(tic0, pri1, dat2, tim3, pri4, pri5, pri6,pri7, cou8),
GetCompanyName(tic0,com15,,),**Add**(pri5,pri13,pri10),**Add**(pri4,pri10,pri1).

5 **YahooAutos**($zip0, $mak1, dat2, yea3, mod4, , , pri7,) :-
GoogleBaseCars(zip0, mak1, , mod4, pri7, , , yea3),
ConvertTime(dat2, , dat10, ,), **GetCurrentTime**(, , dat10,).

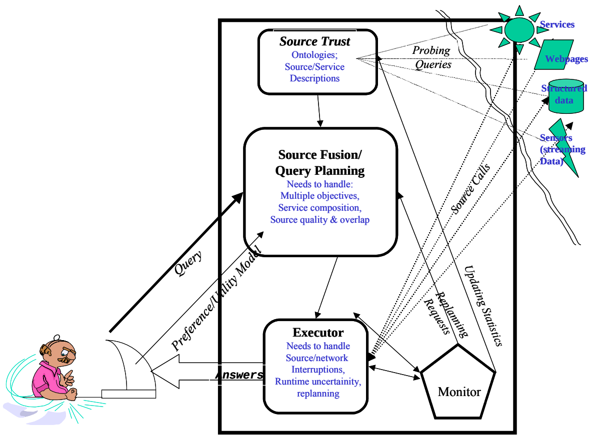
Conclusions

- Assumption: overlap between new & known sources
- Nonetheless, the technique is widely applicable:
  - Redundancy
  - Scope or Completeness
  - Binding Constraints
  - Composed Functionality
  - Access Time



Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Record Linkage Problem

Restaurant Name	Address	City	Phone	Cuisine
Fenix	8358 Sunset Blvd. West	Hollywood	213/848-6677	American
Fenix at the Argyle	8358 Sunset Blvd.	W. Hollywood	213-848-6677	French (new)

L. P. Kaelbling. An architecture for intelligent reactive systems. In Reasoning About Actions and Plans: Proceedings of the 1986 Workshop. Morgan Kaufmann, 1986
Kaelbling, L. P., 1987. An architecture for intelligent reactive systems. In M. P. Georgeff & A. L. Lansky, eds., Reasoning about Actions and Plans, Morgan Kaufmann, Los Altos, CA, 395 410

- Task: **identify syntactically different records that refer to the same entity**
- Common sources of variation: database merges, typographic errors, abbreviations, extraction errors, OCR scanning errors, etc.

General Approach to Record Linkage

1. Identification of candidate pairs (blocking)
    - Comparing all possible record pairs would be computationally wasteful
2. Compute Field Similarity
    - String similarity between individual fields is computed
3. Compute Record Similarity
    - Field similarities are combined into a total record similarity estimate

Blocking – Generating Candidate Matches

Census Data

First Name	*Last Name*	*Phone*	*Zip*
Matt	**Michelson**	**555-5555**	**12345**
Jane	**Jones**	**555-1111**	**12345**
Joe	**Smith**	**555-0011**	**12345**

A.I. Researchers

First Name	*Last Name*	*Phone*	*Zip*
Matthew	**Michelson**	**555-5555**	**12345**
Jim	**Jones**	**555-1111**	**12345**
Joe	**Smeth**	**555-0011**	**12345**

match

match

Approaches to Blocking

- Sort neighborhoods on block keys
  - Hernandez & Stolfo, 1998
- Canopies Method
  - McCallum, Nigam, Ungar, **Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching**, 2000, KDD
- DNF Blocking
  - Bilenko, Kamath, Mooney**, Adaptive Blocking: Learning to Scale Up Record Linkage,** 2006, ICDM
- Blocking Scheme Learning
  - Michelson & Knoblock, **Learning Blocking Schemes for Record Linkage**, 2006, AAAI

Blocking – Multi-pass

- Terminology:
  - Each pass is a “conjunction”
    - (token, first) AND (token, phone)
  - Combine passes to form “disjunction”
    - [(token, last)] OR [(token, first) AND (token, phone)]
  - Disjunctive Normal Form rules
    - form “Blocking Schemes”

Blocking – Generating Candidates

(token, last name) AND (1st letter, first name)

First Name	*Last Name*

< Matt, Michelson>	↔ ✓	**< Matthew, Michelson>**	
< Jane, Jones>	↔ ✗	**< Jim, Jones>**	

First Name	*Last Name*

(token, zip)

First Name	*Last Name*	*Zip*

< Matt, Michelson, 12345 >	↔ ✓	< Matthew, Michelson, 12345 >
< Matt, Michelson, 12345 >	↔ ✗	< Jim, Jones, 12345 >
< Matt, Michelson, 12345 >	↔ ✗	< Joe, Smeth, 12345 >
⋮		⋮

First Name	*Last Name*	*Zip*

Blocking Effectiveness

Reduction Ratio (RR) = $1 - \|C\| / (\|S\| * \| T\|)$

S,T are data sets; C is the set of candidates

Pairs Completeness (PC) = S_m / N_m

S_m = # true matches in candidates,

N_m = # true matches between S and T

Examples: **(token, last name) AND (1^{st} letter, first name)**

RR = $1 - 2/9 \approx 0.78$

PC = $1 / 2 = 0.50$

(token, zip)

RR = $1 - 9/9 = 0.0$

PC = $2 / 2 = 1.0$

How to choose methods and attributes?

- Blocking Goals:
  - Small number of candidates (High RR)
  - Don’t leave any true matches behind! (High PC)
- Previous approaches:
  - Ad-hoc by researchers or domain experts
- Learning Approach:
  - BSL – “Blocking Scheme Learner”
    - modified Sequential Covering Algorithm

Learning Schemes – Intuition

- Learn restrictive conjunctions
  - partition the space → minimize False Positives
- Union restrictive conjunctions
  - Cover all training matches
  - Since minimized FPs, conjunctions should not contribute many FPs to the disjunction

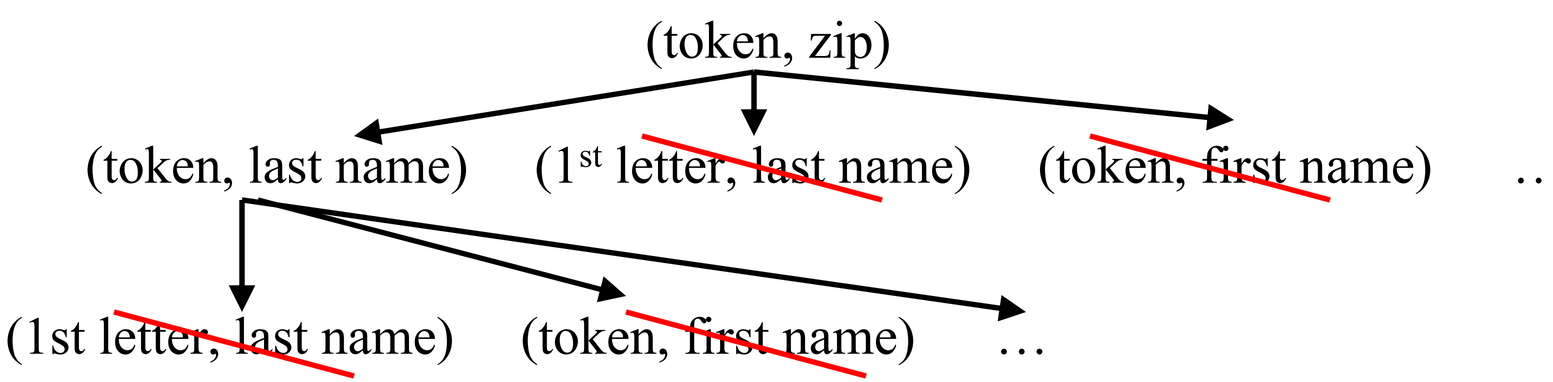
SCA: propositional rules

- Multi-pass blocking = disjunction of conjunctions
- Learn conjunctions and union them together!
- Cover all training matches to maximize PC

```
SEQUENTIAL-COVERING( class, attributes, examples, threshold)
LearnedRules ← {}
Rule ← LEARN-ONE-RULE(class, attributes, examples)
While examples left to cover, do
        LearnedRules ← LearnedRules U Rule
        Examples ← Examples – {Examples covered by Rule}
        Rule ← LEARN-ONE-RULE(class, attributes, examples)
        If Rule contains any previously learned rules, remove them
Return LearnedRules
```

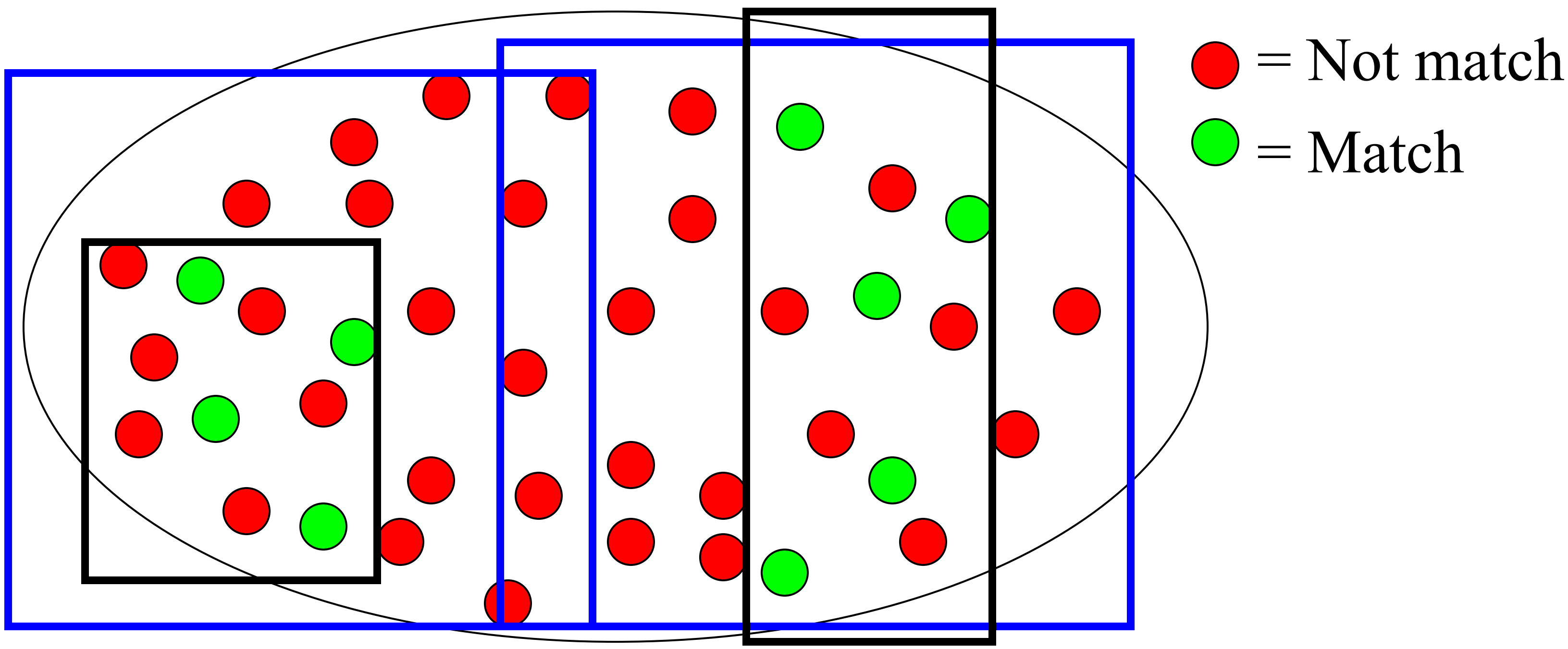
Learn-One-Rule

- Learn conjunction that maximizes RR
- General-to-specific beam search
  - Keep adding/intersecting (attribute, method) pairs
    - Until can't improve RR
    - Must satisfy minimum PC



Example to clear things up!

Space of training examples



Rule 1 :- (zip|token) & (first|token)

Final Rule :- [(zip|token) & (first|token)] UNION [(last|1st Letter) & (first|1st letter)]

Rule 2 :- (last|1st Letter) & (first|1st Letter)

Experiments

Cars	RR	PC
HFM	47.92	99.97*
BSL	99.86	99.92*
BSL (10%)	99.87	99.88

HFM = ({token, make} ∩ {token, year} ∩ {token, trim})

U ({1st letter, make} ∩ {1st letter, year} ∩ {1st letter, trim})

U ({synonym, trim})

B S L = ({token, model} ∩ {token, year} ∩ {token, trim})

U ({token, model} ∩ {token, year} ∩ {synonym, trim})

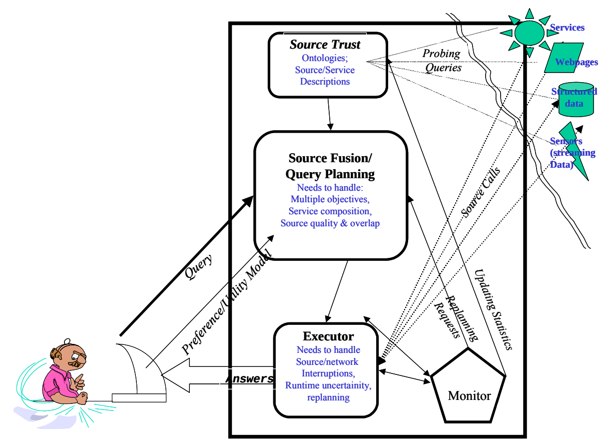
Census	RR	PC
Best 5 Winkler	99.52	99.16
Adaptive Filtering	99.9	92.7
BSL	98.12	99.85
BSL (10%)	99.50	99.13

Restaurants	RR	PC
Marlin	55.35	100.00
BSL	99.26	98.16
BSL (10%)	99.57	93.48

* = NOT statistically significant.

Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Field Matching Approaches

- Expert-system rules
  - Manually written (e.g., Lexus Nexus)
- Token similarity
  - Used in Whirl [Cohen, ACM TOIS 2000]
- String similarity
  - Used in Marlin [Bilenko & Moody, KDD 2003]
- Domain-specific transformations
  - Used in Active Atlas [Tejada, Knoblock & Minton, KDD 2002]
- Heterogeneous Field Matching
  - Used in HFM [Minton, et al., ICDM 2005]

Token Similarity [Cohen, 1998]

- Idea: Evaluate the similarity of records via textual similarity
  - Used in Whirl (Cohen 1998)
- Any string can be treated as a *bag of tokens* .
  - *“8358 Sunset Blvd”* ► *{8358, Sunset, Blvd}*
- Follows the same approach used by classical IR algorithms (including web search engines)
  - “stemming” is applied to each entry
    - E.g. “Joe’s Diner” -> “Joe [‘s] Diner”
  - Entries are compared by counting the number of words in common
  - Infrequent words weighted more heavily by TF/IDF metric = Term Frequency / Inverse Document Frequency

Sequence-based String Metrics: String Edit Distance [Levenshtein, 1966]

- Minimum number of character *deletions*, *insertions*, or *substitutions* needed to make two strings equivalent.
  - “misspell” to “mispell” is distance 1 (*‘delete s’*)
  - “misspell” to “mistell” is distance 2 (*‘delete s’, ‘substitute p with t’* OR *‘substitute s with t’, ‘delete p’*)
  - “misspell” to “misspelling” is distance 3 (‘*insert i’, ‘insert n’, ‘insert g’*)
- Can be computed efficiently using dynamic programming in O(*mn*) time where *m* and *n* are the lengths of the two strings being compared.
- Unit cost is typically assigned to individual edit operations, but individual costs can be used.

String Edit Distance with Affine Gaps [Gotoh,1982]

- Cost of gaps formed by *contiguous deletions/insertions* should be lower than the cost of multiple non-contiguous operators.
  - Distance from “misspell” to “misspelling” is <3.
- Affine model for gap cost: $\mathrm{cost}(gap) = s + e|gap|,\ e < s$
- Edit distance with affine gaps is more flexible since it is less susceptible to sequences of insertions/deletions that are frequent in natural language text (e.g.*’Street’* vs. ‘*Str*’).

Learnable Edit Distance with Affine Gaps

[Bilenko & Moody, 2003]

- Motivation:

  Significance of edit operations depends on a particular domain

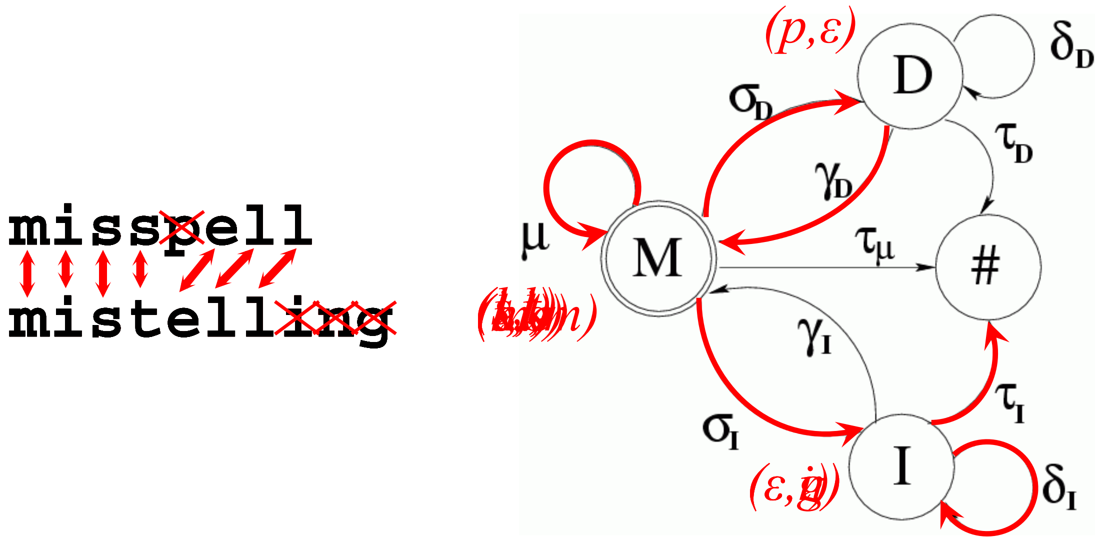
  – *Substitute '/' with '-'* insignificant for phone numbers.

  – *Delete 'Q'* significant for names.

  – Gap start/extension costs vary: sequence deletion is common for addresses (*'Street'* ►*'Str'*), uncommon for zip codes.

- Using individual weights for edit operations, as well as learning gap operation costs allows adapting to a particular field domain.

Learnable Edit Distance with Affine Gaps – the Generative Model



- Matching/substituted pairs of characters are generated in state *M.*
- Deleted/inserted characters that form gaps are generated in states *D* and *I.*
- Special termination state "#" ends the alignment of two strings.
- Similar to pairwise alignment HMMs used in bioinformatics [Durbin *et al.* '98]

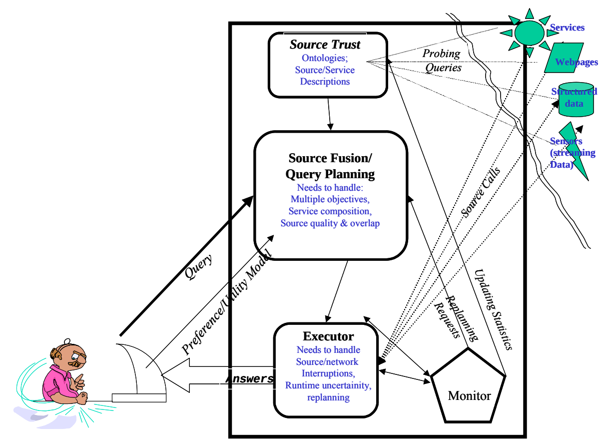
Learnable Edit Distance with Affine Gaps: Training

- Given a corpus of *matched* string pairs, the model is trained using Expectation-Maximization.
- The model parameters take on values that result in high probability of producing duplicate strings.
  - Frequent edit operations and typos have *high* probability.
  - Rare edit operations have *low* probability.
  - Gap parameters take on values that are optimal for duplicate strings in the training corpus.
- Once trained, distance between any two strings is estimated as

  the posterior probability of generating the most likely alignment between the strings as a sequence of edit operations.
- Distance computation is performed in a simple dynamic programming algorithm.

Overview

- Motivation & Models for Information Integration [30]
  - Models for integration
  - Semantic Web
- Getting Data into structured format [30]
  - Wrapper Construction
  - Information Extraction
- Getting Sources into alignment [30]
  - Schema Mapping
  - Source Modeling
- Getting Data into alignment [30]
  - Blocking
  - Field Matching
  - Record Linkage
- Processing Queries [45]
  - Autonomous sources; data uncertainty..
  - Plan Execution
- Wrapup [15]



Combining String Similarity Across Fields

- Some fields are more indicative of record similarity than others:
  - For addresses, *street address* similarity is more important than *city* similarity.
  - For bibliographic citations, *author* or *title* similarity are more important than *venue* (i.e. conference or journal name) similarity.
- Field similarities should be weighted when combined to determine record similarity.
- Weights can be learned using a learning algorithm [Cohen & Richman '02], [Sarawagi & Bhamidipaty '02], [Tejada *et. al*. '02].

Record Matching Approaches

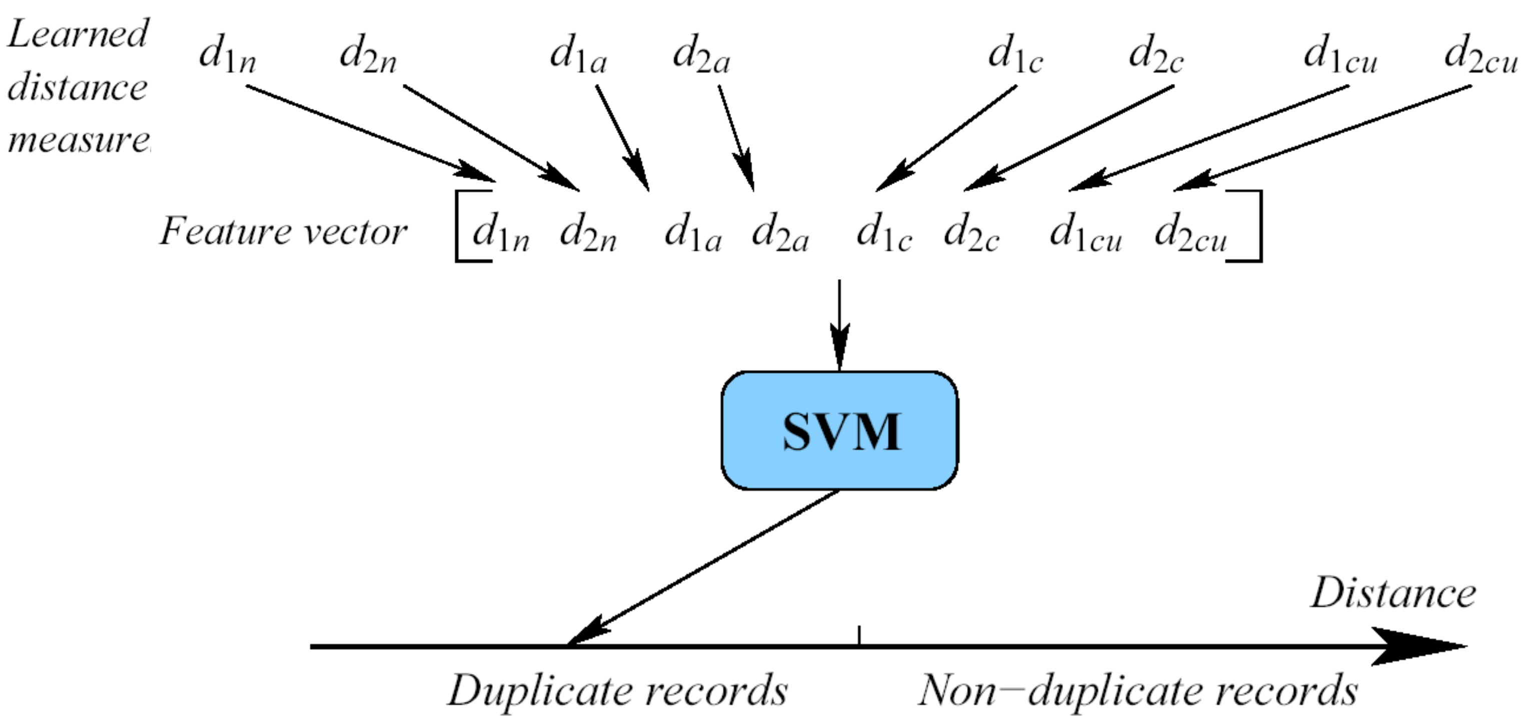
- Unsupervised Record linkage [Newcombe et al. ’59; Fellegi & Sunter ’69; Winkler ’94, ’99, ‘02]
- Merge/purge [Hernandez & Stolfo ’95]
- Database hardening [Cohen et al. ’00]
- Learning Decision Trees [Tejada, Knoblock & Minton, KDD 2002, Sarawagi & Bhamidipaty KDD 2002]
- Support Vector Machines (SVM) [Bilenko & Moody, KDD 2003]
- Object consolidation [Michalowski et al. ’03]

SVM Learned Record Similarity

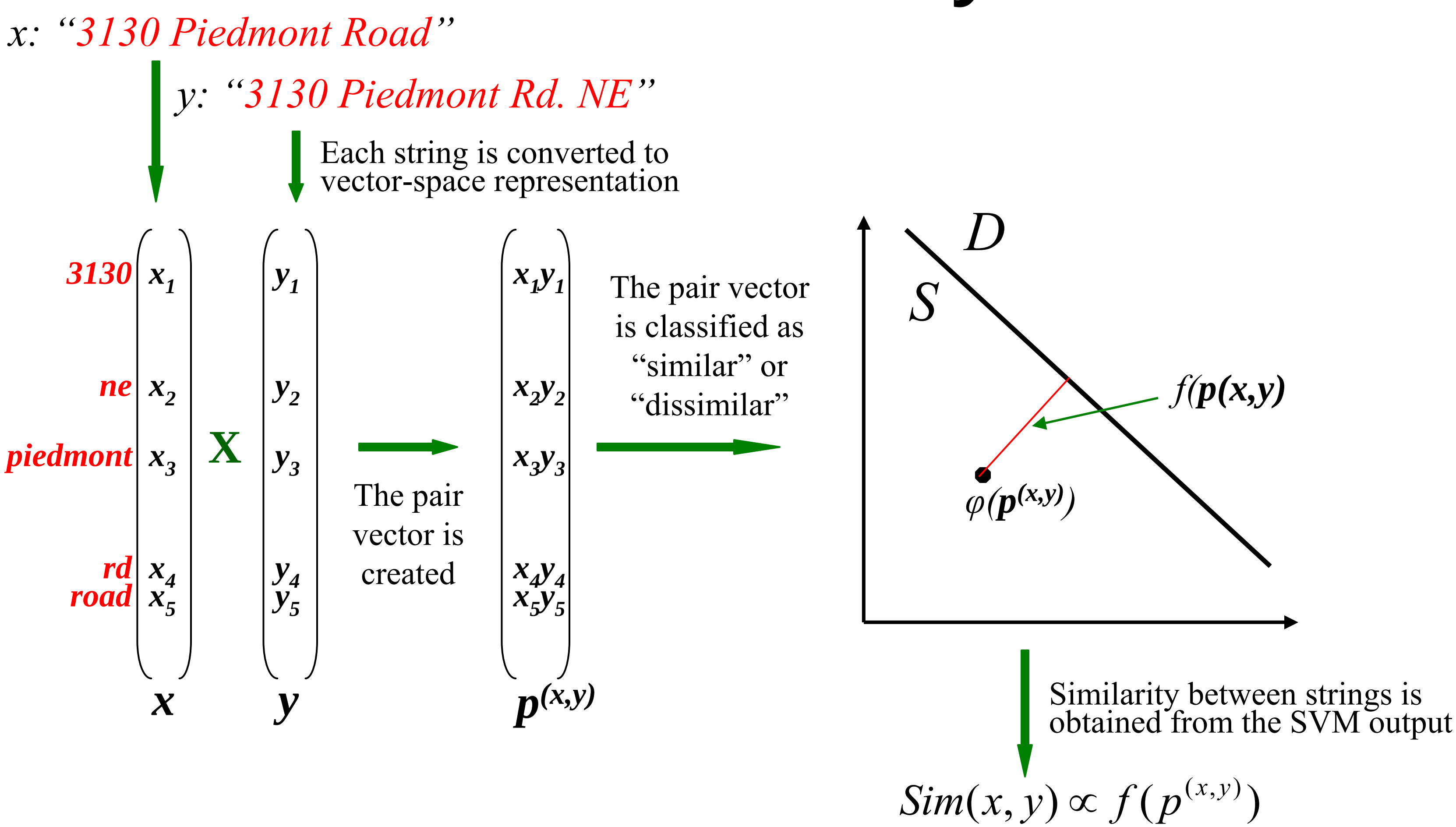
- String similarities for each field are used as components of a feature vector for a pair of records.
- SVM is trained on labeled feature vectors to discriminate duplicate from non-duplicate pairs.
- Record similarity is based on the distance of the feature vector from the separating hyperplane.

Learning Record Similarity (cont.)

Name	***Address***	***City***	***Cuisine***
Fenix	8358 Sunset Blvd. West	Hollywood	American
Fenix at the Argyle	8358 Sunset Blvd.	W. Hollywood	French (new)



Learnable Vector-space Similarity

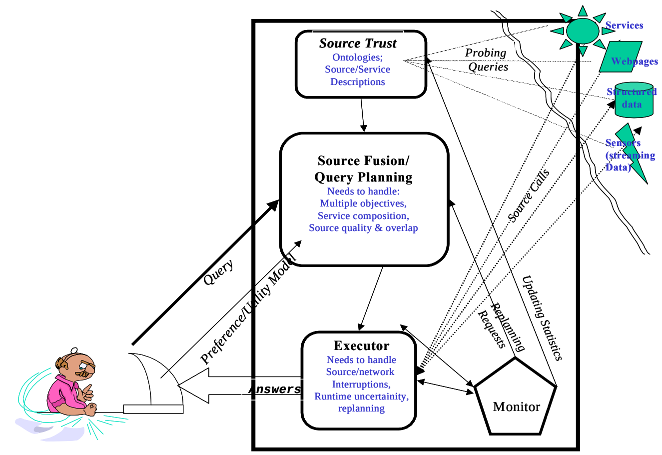


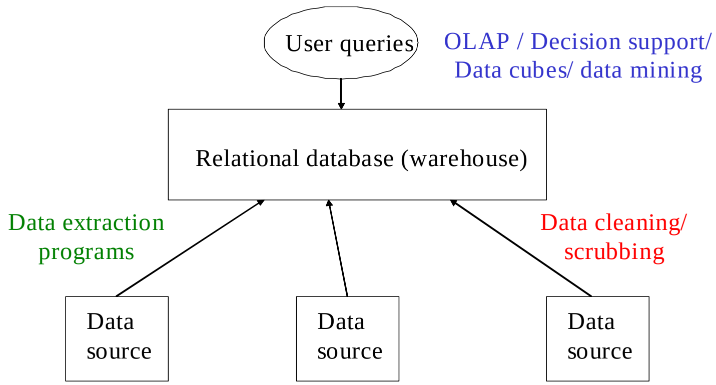
Conclusions

- Technical choices in record linkage:
  - Approach to blocking
  - Approach to field matching
  - Approach to record matching
- Learning approaches have the advantage of being able to
  - Adapt to specific application domains
  - Learn which fields are important
  - Learn the most appropriate transformations
- Optimal classifier choice is sensitive to the domain and the amount of available training data.

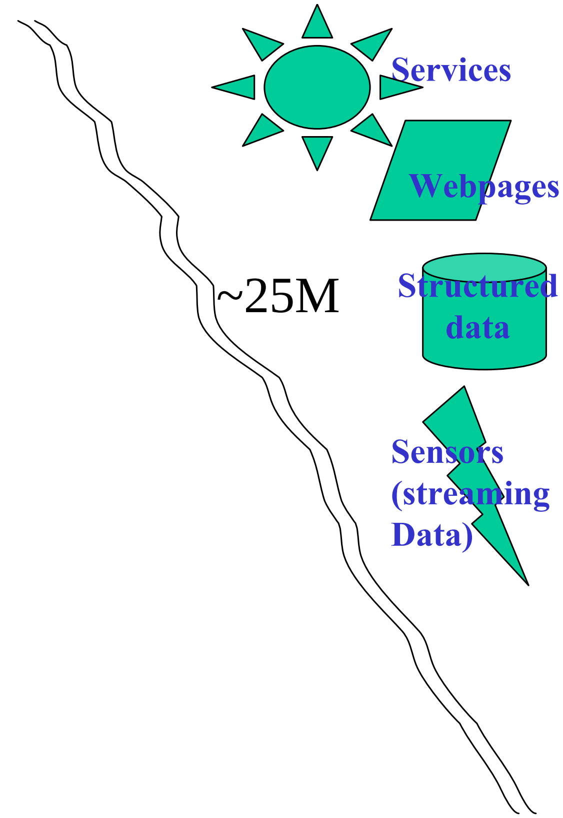
Overview

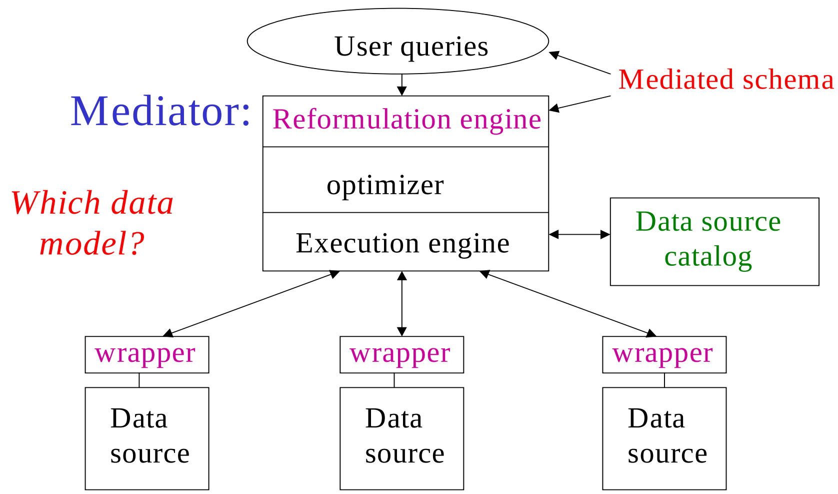
- Motivation & Models for Information Integration [30]
- Getting Data into structured format [30]
- Getting Sources into alignment [30]
- Getting Data into alignment [30]
- Processing Queries [45]
- Wrapup [15]



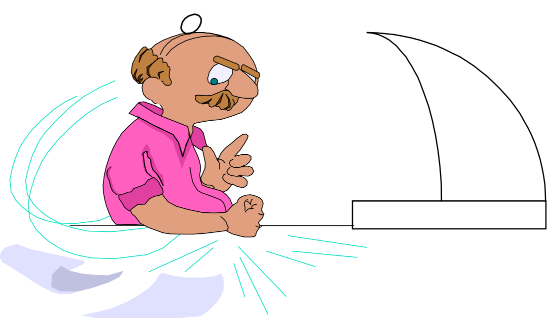


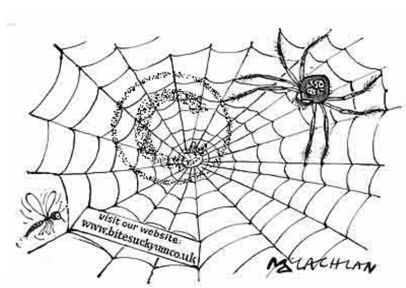
--Warehouse Model--
--Get all the data and
put into a local DB
--Support structured
queries on the
warehouse





--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results





--Search Model--
--Materialize
the pages
--crawl &index them
--include them in
search results

Extracting Information
Aligning Sources
Aligning Data

Kambhampati & Knoblock

Query Processing Challenges

Supporting Imprecision/ Incompleteness/Uncertainty

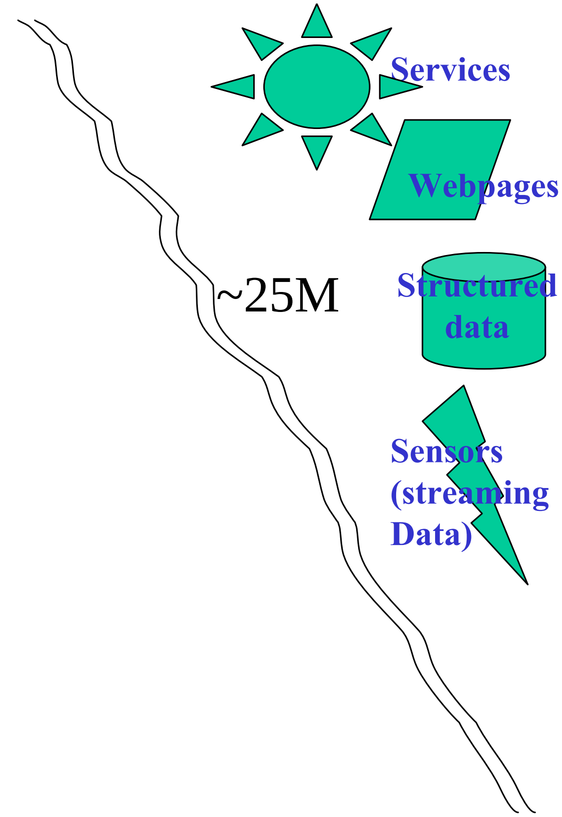
Query reformulation
Optimizing Access to Sources

Indexing with Structure

--Warehouse Model--
--Get all the data and put into a local DB
--Support structured queries on the warehouse

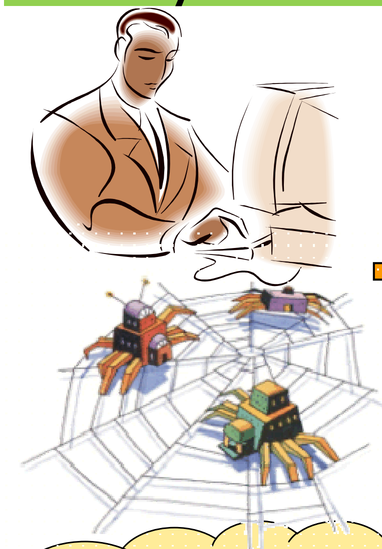
--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results

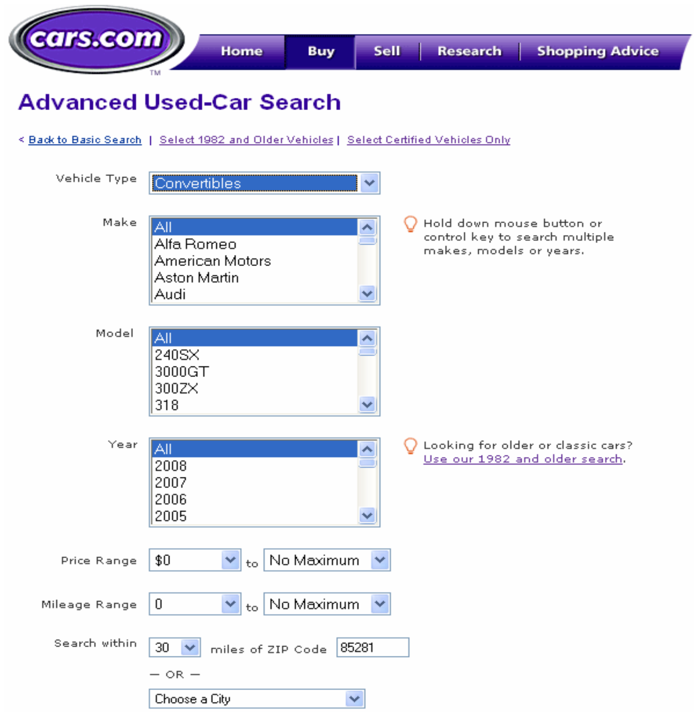
--Search Model--
--Materialize the pages
--crawl &index them
--include them in search results

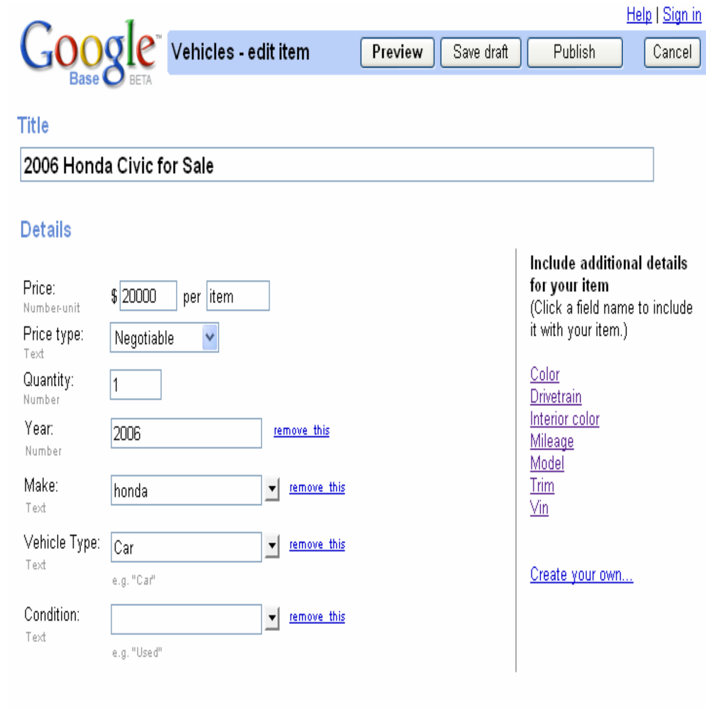


Incompleteness in Web databases

Populated by Lay Users







Automated Extraction

Website	# of attributes	# of tuples	incomplete tuples	body style	engine
autotrader.com	13	25127	33.67%	3.6%	8.1%
carsdirect.com	14	32564	98.74%	55.7%	55.8%

Imprecise Queries ?

Want a ‘sedan’ priced around $7000

A Feasible Query

Make =“Toyota”,
Model=“Camry”,
Price ≤ $7000

Toyota	Camry	$7000	1999
Toyota	Camry	$7000	2001
Toyota	Camry	$6700	2000
Toyota	Camry	$6500	1998
………			

What about the price of a Honda Accord?

Is there a Camry for $7100?

Solution: Support Imprecise Queries

Digression: DB ←→ IR

Databases

User **knows** what she wants

User query **completely** expresses the need

Answers **exactly** matching query constraints

IR Systems

- User **has an idea** of what she wants
- User query captures the need to **some degree**
- Answers **ranked** by degree of relevance

Can see the challenges as "Structured IR" or "Semi-structured DB"

Imprecision & Incompleteness

Imprecise Queries

User's needs are not clearly defined hence:

- **Queries may be too general**
- **Queries may be too specific**

Incomplete Data

Databases are often populated by:

- **Lay users entering data**
- **Automated extraction**

Relevance Function

$$\mathcal{ER}(\hat{t}|Q,U,D) = \sum_{t \in C(\hat{t})} \mathcal{R}(t|Q,U)\mathcal{P}(t|\hat{t},D)$$

Density Function

General Solution: **"Expected Relevance Ranking"**

Challenge: Automated & Non-intrusive assessment of Relevance and Density functions

However, how can we retrieve similar/ incomplete tuples in the first place?

Challenge: Rewriting a user's query to retrieve highly relevant Similar/ Incomplete tuples

Once the similar/incomplete tuples have been retrieved, why should users believe them?

Challenge: Provide explanations for the uncertain answers in order to gain the user's trust

Retrieving Relevant Answers via Query Rewriting

Problem:

How to rewrite a query to retrieve answers which are highly relevant to the user?

Given a query Q:(Model=Civic) retrieve all the relevant tuples

1. Retrieve certain answers namely tuples t_1 and t_6
2. Given an AFD, rewrite the query using the determining set attributes in order to retrieve possible answers

$$\text{AFD } \{Make, BodyStyle\} \rightsquigarrow Model$$

   a) Q_1': Make=Honda Λ Body Style=coupe

   b) Q_2': Make=Honda Λ Body Style=sedan

Thus we retrieve:

- ✓ Certain Answers
- ✓ Incomplete Answers
- ✓ Similar Answers

ID	Make	Model	Year	Color	Body Style
1	Honda	Civic	2000	red	coupe
2	Honda	Accord	2004	blue	sedan
3	Toyota	Camry	2001	silver	sedan
4	Honda	NULL	2004	black	coupe
5	BMW	3-series	2001	blue	convt
6	Honda	Civic	2004	green	sedan
7	Honda	NULL	2000	white	sedan
8	Honda	Prelude	1999	blue	coupe

Case Study: Query Rewriting in QPIAD

Given a query Q:(Body style=Convt) retrieve all relevant tuples

Id	Make	Model	Year	Body
1	Audi	A4	2001	Convt
2	BMW	Z4	2002	Convt
3	Porsche	Boxster	2005	Convt
4	BMW	Z4	2003	Null
5	Honda	Civic	2004	Null
6	Toyota	Camry	2002	Sedan
7	Audi	A4	2006	Null

Base Result Set

Id	Make	Model	Year	Body
1	Audi	A4	2001	Convt
2	BMW	Z4	2002	Convt
3	Porsche	Boxster	2005	Convt

AFD: Model~> Body style

Rewritten queries

Q_1': Model=A4

Q_2': Model=Z4

Q_3': Model=Boxster

Re-order queries based on Estimated Precision

Ranked Relevant Uncertain Answers

Id	Make	Model	Year	Body	Confidence
4	BMW	Z4	2003	Null	0.7
7	Audi	A4	2006	Null	0.3

We can select top K rewritten queries using F-measure

F-Measure = $(1+\alpha)*P*R/(\alpha*P+R)$

P – Estimated Precision

R – Estimated Recall based on P and Estimated Selectivity

Learning Statistics to support Ranking & Rewriting

- Learning attribute correlations by Approximate Functional Dependency(AFD) and Approximate Key(AKey)

Determining Set(Y) = dtrSet(Y)

Sample Database → TANE → Prune based on AKEY → AFDs (X~>Y)+ confidence

- Learning value distributions using Naïve Bayes Classifiers(NBC)

Determining Set(A_m) → Feature Selection → Learn NBC classifiers with m-estimates → Estimated Precision = P(A_m=v_m|dtrSet(A_m))

- Learning Selectivity Estimates of Rewritten Queries(Q'_{Sel}) based on:
  - Selectivity of rewritten query issued on sample
  - Ratio of original database size over sample
  - Percentage of incomplete tuples while creating sample

Explaining Results to Users

Problem:

How to gain users trust when showing them similar/incomplete tuples?

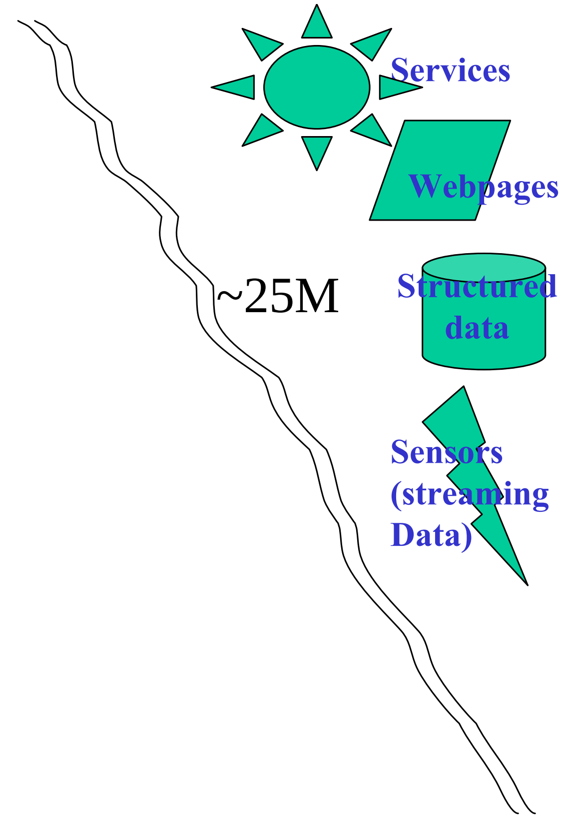
Query Results for query

Make like honda and *Model* like civic and *Year* like 2001

Make	Model	Year	Price	Mileage	Location	Color	Relevant	Explanation
honda	civic	2001	16662	58977	Tempe	blue	☐	
honda	civic	2001	18610	16667	Mesa	red	☐	
honda	civic	2001	15994	48123	Chandler	silver	☐	
?	civic	2001	13490	58977	Phoenix	silver	☐	This car is 100% likely to have make=honda given that its model=civic
honda	civic	2003	17490	16667	Phoenix	gray	☐	Cars having year=2003 are 80% similar to cars having year=2001
honda	accord	2001	15994	48123	Gilbert	silver	☐	Cars having model=accord are 78% similar to cars having model=civic given that 78% of users who looked at civic also looked at accord
honda	?	2001	14995	32533	Mesa	black	☐	This car is 73% likely to have model=civic given that its make=honda, year=2001, and color=black
honda	?	2001	15990	43137	Tempe	silver	☐	This car is 32% likely to have model=accord given that its make=honda, year=2001, and color=silver and 78% of users who looked at civic also looked at accord

QUIC Demo at rakaposhi.eas.asu.edu/quic

Query Processing Challenges



Supporting Imprecision/ Incompleteness/Uncertainty

Query reformulation
Optimizing Access to Sources

Indexing with Structure

--Warehouse Model--
--Get all the data and put into a local DB
--Support structured queries on the warehouse

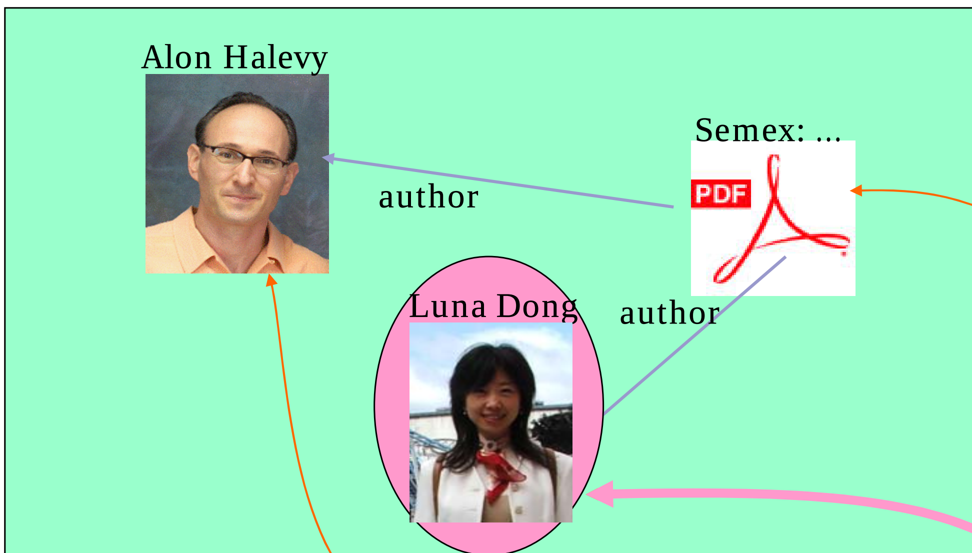
--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results

--Search Model--
--Materialize the pages
--crawl &index them
--include them in search results

Kambhampati & Knoblock

Web Search Model: Keyword Queries with Inverted Lists

How about queries such as “FirstName Dong” or “Author Dong”



Departmental Database

StuID	lastName	firstName	…
1000001	Xin	Dong	…
…	…	…	…

Inverted List

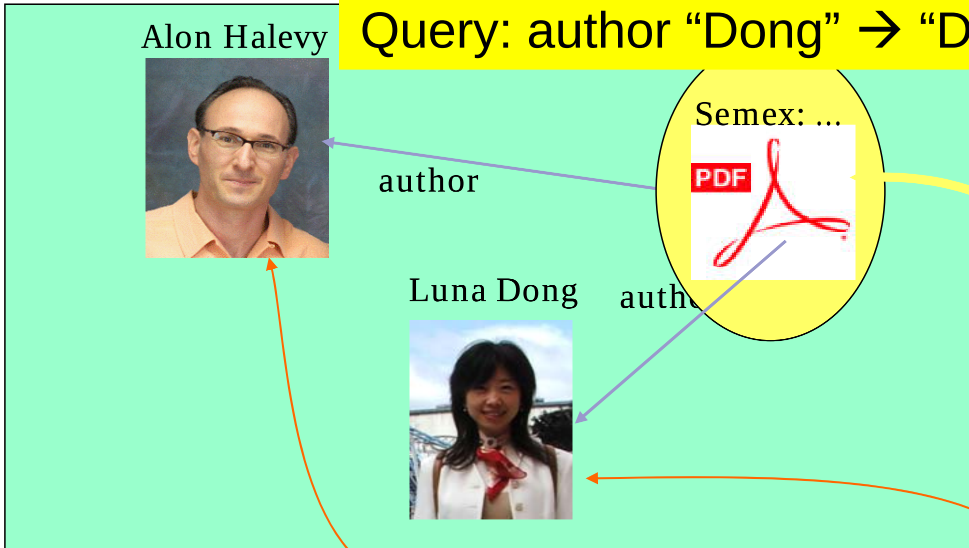
Alon	1			
Dong		1		1
Halevy	1			
Luna		1		
Semex			1	
Xin				1

Query: Dong

[Slide courtesy Xin Dong]

Web Search Model: Structure-aware Keyword Queries (with extended Inverted Indices)

Query: author “Dong” → “Dong/author/”



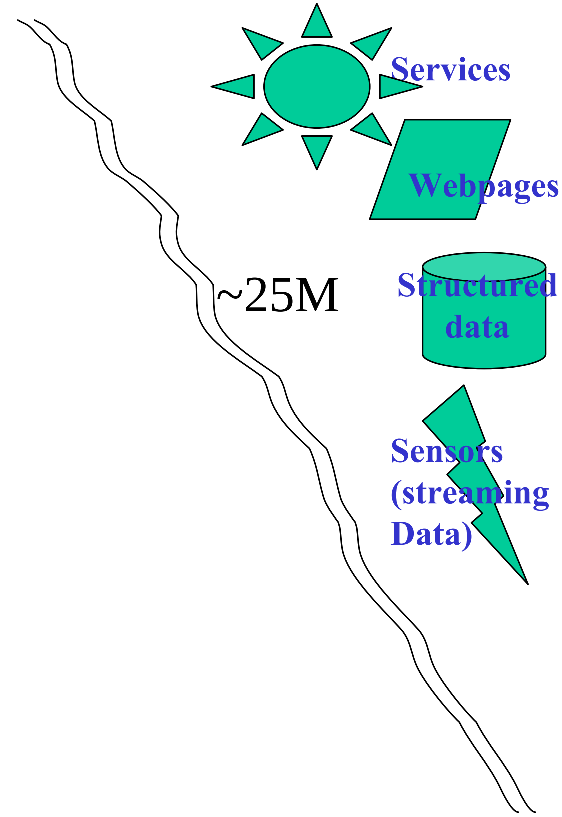
Departmental Database

StuID	LastName	FirstName	…
1000001	Xin	Dong	…
…	…	…	…

Inverted List (extended with attribute labels & association labels)

Alon/author/			**1**	
Alon/name/	1			
Dong/author/			**1**	
Dong/name/		1		
Dong/name/firstName/				1
Halevy/name/	1			
Luna/name/		1		
Luna/auhor			1	
Semex/title/			1	

Query Processing Challenges



--Warehouse Model--
--Get all the data and put into a local DB
--Support structured queries on the warehouse

Supporting Imprecision/ Incompleteness/Uncertainty

--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results

Query reformulation
Optimizing Access to Sources

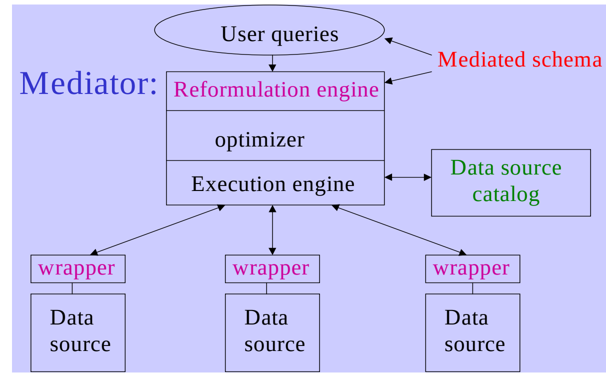
Indexing with Structure

--Search Model--
--Materialize the pages
--crawl &index them
--include them in search results

Kambhampati & Knoblock

Desiderata for Relating Source-Mediator Schemas

- Expressive power: distinguish between sources with closely related data. Hence, be able to prune access to irrelevant sources.
- Easy addition: make it easy to add new data sources.
- Reformulation: be able to reformulate a user query into a query on the sources efficiently and effectively.
- Nonlossy: be able to handle all queries that can be answered by directly accessing the sources



Reformulation

- Given:
  - A query Q posed over the mediated schema
  - Descriptions of the data sources
- Find:
  - A query Q' over the data source relations, such that:
    - Q' provides only *correct answers* to Q, and
    - Q' provides *all* possible answers to Q given the sources.

Differences minor for data aggregation…

Approaches for relating source & Mediator Schemas

- Global-as-view (GAV): express the mediated schema relations as a set of views over the data source relations
- Local-as-view (LAV): express the source relations as views over the mediated schema.
- Can be combined…?

"View" Refresher

```
CREATE VIEW  Seattle-view  AS

        SELECT  buyer, seller, product, store
        FROM    Person, Purchase
        WHERE   Person.city = "Seattle"    AND
                Person.name = Purchase.buyer
```

We can later use the views:

```
        SELECT   name, store
        FROM     Seattle-view, Product
        WHERE    Seattle-view.product = Product.name  AND
                 Product.category = "shoes"
```

Virtual vs Materialized

Let's compare them in a movie Database integration scenario..

Global-as-View

Mediated schema:

Movie(title, dir, year, genre),

Schedule(cinema, title, time).

Express mediator schema relations as views over source relations

[S1(title,dir,year,genre)]

[S2(title, dir,year,genre)]

[S3(title,dir),

S4(title,year,genre)]

Global-as-View

Mediated schema:

Movie(title, dir, year, genre),

Schedule(cinema, title, time).

Express mediator schema relations as views over source relations

Create View Movie AS

select * from S1   [S1(title,dir,year,genre)]

union

select * from S2   [S2(title, dir,year,genre)]

union   [S3(title,dir), S4(title,year,genre)]

select S3.title, S3.dir, S4.year, S4.genre

from S3, S4

where S3.title=S4.title

Mediator schema relations are Virtual views on source relations

Local-as-View: example 1

Mediated schema:

Movie(title, dir, year, genre),

Schedule(cinema, title, time).

Express source schema relations as views over mediator relations

Create Source S1 AS

select * from Movie

S1(title,dir,year,genre)

Create Source S3 AS

select title, dir from Movie

S3(title,dir)

Create Source S5 AS

select title, dir, year

from Movie

where year > 1960 AND genre="Comedy"

S5(title,dir,year), year >1960

Sources are "materialized views" of mediator schema

GAV vs. LAV

Mediated schema:

Movie(title, dir, year, genre),

Schedule(cinema, title, time).

Source S4: S4(cinema, genre)

```
Create View Movie AS
  select NULL, NULL, NULL, genre
  from S4
Create View Schedule AS
  select cinema, NULL, NULL
  from S4.
```

But what if we want to find which cinemas are playing comedies?

Lossy mediation

```
Create Source S4
  select cinema, genre
  from Movie m, Schedule s
  where m.title=s.title
```

Now if we want to find which cinemas are playing comedies, there is hope!

GAV vs. LAV

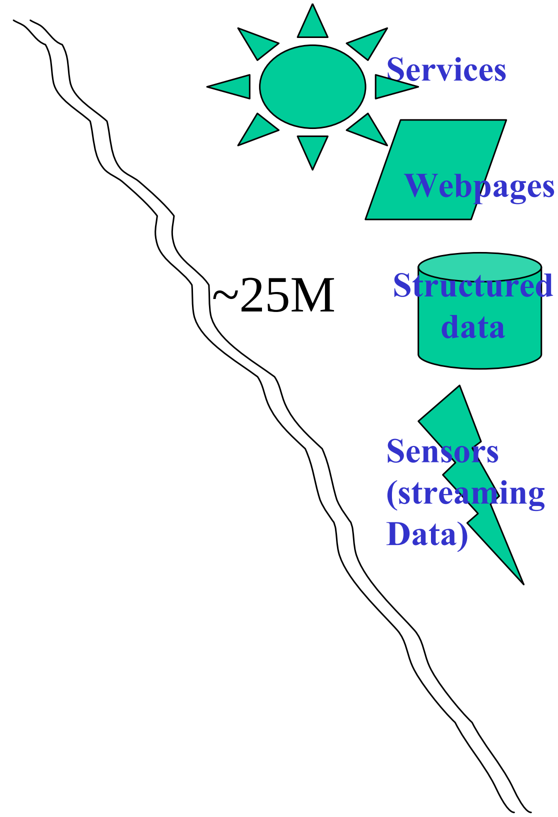
GAV

- Not modular
  - Addition of new sources changes the mediated schema
- Can be awkward to write mediated schema without loss of information
- Query reformulation easy
  - *reduces to view unfolding (polynomial)*
  - Can build hierarchies of mediated schemas
- Best when
  - Few, stable, data sources
  - well-known to the mediator (e.g. corporate

LAV

- Modular--adding new sources is easy
- Very flexible--power of the entire query language available to describe sources
- Reformulation is hard
  - Involves answering queries only using views (can be intractable—see below)
- Best when
  - Many, relatively unknown data sources
  - possibility of addition/deletion of sources

Query Processing Challenges



--Warehouse Model--
--Get all the data and put into a local DB
--Support structured queries on the warehouse

Supporting Imprecision/ Incompleteness/Uncertainty

--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results

Query reformulation
Optimizing Access to Sources

Indexing with Structure

--Search Model--
--Materialize the pages
--crawl &index them
--include them in search results

What to Optimize?

- Traditional DB optimizers compare candidate plans purely in terms of the time they take to produce *all* answers to a query.
- In Integration scenarios, the optimization is "*multi-objective*"
  - Total time of execution
  - Cost to first few tuples
    - Often, the users are happier with plans that give first tuples faster
  - Coverage of the plan
    - Full coverage is no longer an iron-clad requirement
      - Too many relevant sources, Uncontrolled overlap between the sources
    - Can't call them all!
  - (Robustness,
  - Access premiums…)

Source Selection

- All sources are exporting fragments of the same relation **R**
  - E.g. Employment opps; bibliography records; item/price records etc
  - The fragment of R exported by a source may have fewer columns and/or fewer rows
- The main issue in DA is "Source Selection"
  - Given a query q, which source(s) should be selected and in what order
- Objective: Call the least number of sources that will give most number of high-quality tuples in the least amount of time
  - Decision version: Call k sources that ….
  - Quality of tuples– may be domain specific (e.g. give lowest price records) or domain independent (e.g. give tuples with fewest null values)

Issues affecting Source Selection in

- Source Overlap
  - In most cases you want to *avoid* calling overlapping sources
  - …but in some cases you want to *call overlapping sources*
    - E.g. to get as much information about a tuple as possible; to get the lowest priced tuple etc.
- Source latency
  - You want to call sources that are likely to respond fast
- Source quality
  - You want to call sources that have high quality data
    - Domain independent: E.g. High density (fewer null values)
    - Domain specific E.g. sources having lower cost books
- Source "consistency"?
  - Exports data that is error free

Learning Source Statistics

- Coverage, overlap, latency, density and quality statistics about sources are not likely to be exported by sources!
  - Need to learn them
- Most of the statistics are source and *query* specific
  - Coverage and Overlap of a source may depend on the query
  - Latency may depend on the query
  - Density may depend on the query
- Statistics can be learned in a qualitative or quantitative way
    - LCW vs. coverage/overlap statistics
    - Feasible access patterns vs. binding pattern specific latency statistics
  - Quantitative is more general and amenable to learning
- Too costly to learn statistics w.r.t. each specific query
  - Challenge: Find right type of query classes with respect to which statistics are learned
    - Query class definition may depend on the type of statistics
- Since sources, user population and network are all changing, statistics need to be *maintained* (through incremental changes)

Case Study: Learning Source Overlap

- Often, sources on the Internet have overlapping contents
  - The overlap is *not* centrally managed (unlike DDBMS—data replication etc.)
- Reasoning about overlap is important for plan optimality
  - We cannot possibly call all potentially relevant sources!
- Qns: How do we characterize, get and exploit source overlap?
  - Qualitative approaches (LCW statements)
  - Quantitative approaches (Coverage/Overlap statistics)

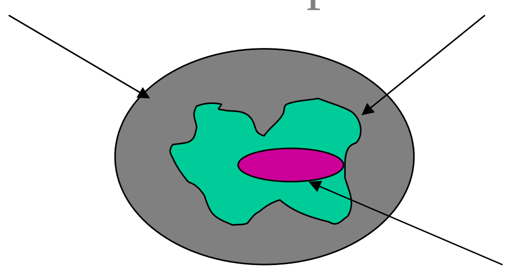
Local Completeness Information

- If sources are incomplete, we need to look at each one of them.
- Often, sources are *locally complete*.
- Movie(title, director, year) complete for years after 1960, or for American directors.
- Question: given a set of local completeness statements, is a query Q' a complete answer to Q?

Problems:

1. Sources may not be interested in giving these!
   →Need to learn
   →hard to learn!

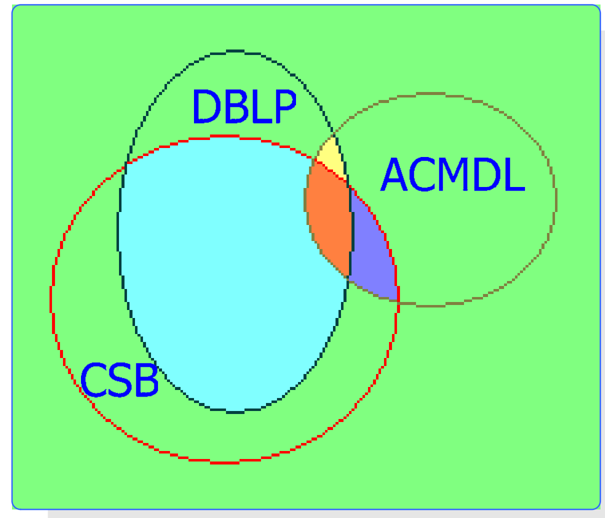
2. Even if sources are willing to give, there may not be any "big enough" LCWs
   Saying "I definitely have the car with vehicle ID XXX is useless



Quantitative ways of modeling inter-source overlap

Coverage: probability that a random answer tuple for query Q belongs to source S. Noted as P(S|Q).

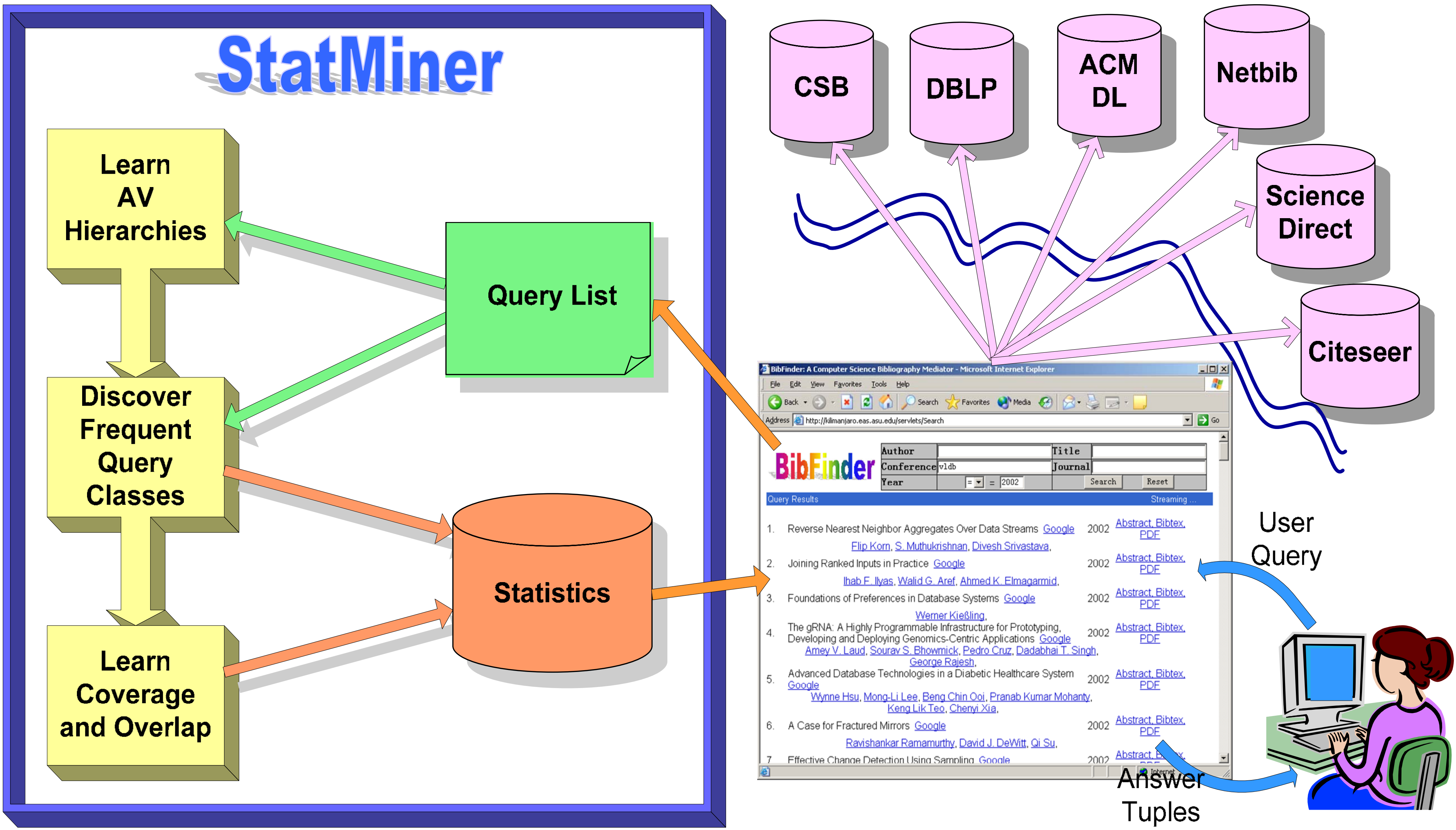
Overlap: Degree to which sources contain the same answer tuples for query Q. Noted as $P(S_1 \wedge S_2 \wedge \ldots \wedge S_k | Q)$.



$$P(DBLP \mid Q, CSB) = P(DBLP \mid Q) - P(CSB \wedge DBLP \mid Q)$$

- Need *coverage* and *overlap* statistics to figure out what sources are most relevant for every possible query!
  - Who gives the statistics?

BibFinder/StatMiner



Query List & Raw Statistics

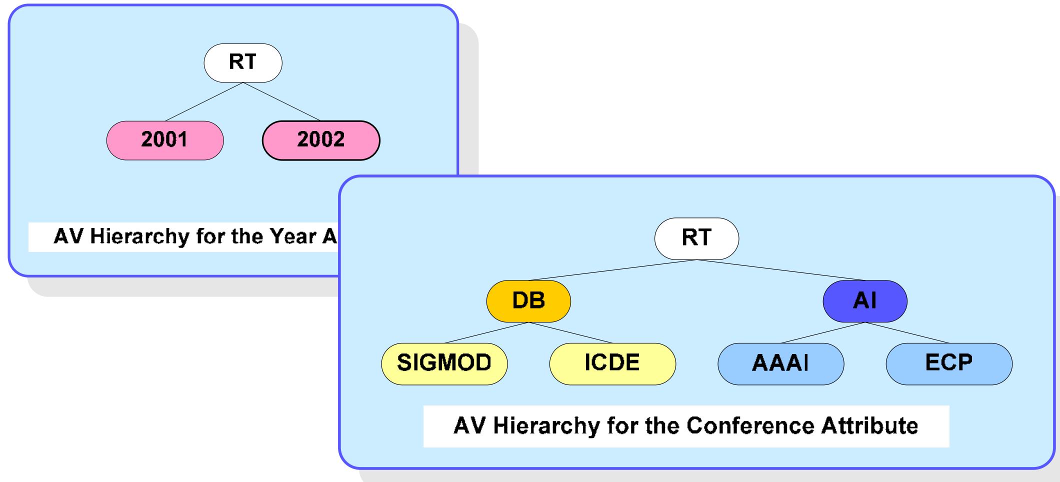
Query	Frequency	Distinctive Answers	Overlap (Coverage)	
Author="andy king"	106	46	DBLP	35
			CSB	23
			CSB, DBLP	12
			DBLP, Science	3
			Science	3
			CSB, DBLP, Science	1
			CSB, Science	1
Author="fayyad" Title="data mining"	1	27	CSB	16
			DBLP	16
			CSB, DBLP	7
			ACMdl	5
			ACMdl, CSB	3
			ACMdl, DBLP	3
			ACMdl, CSB, DBLP	2
			Science	1

Query List: the mediator maintains an XML log of all user queries, along with their access frequency, number of total distinct answers obtained, and number of answers from each source set which has answers for the query.

Given the query list, we can compute the raw statistics for each query: $P(S1..Sk|q)$

AV Hierarchies and Query Classes

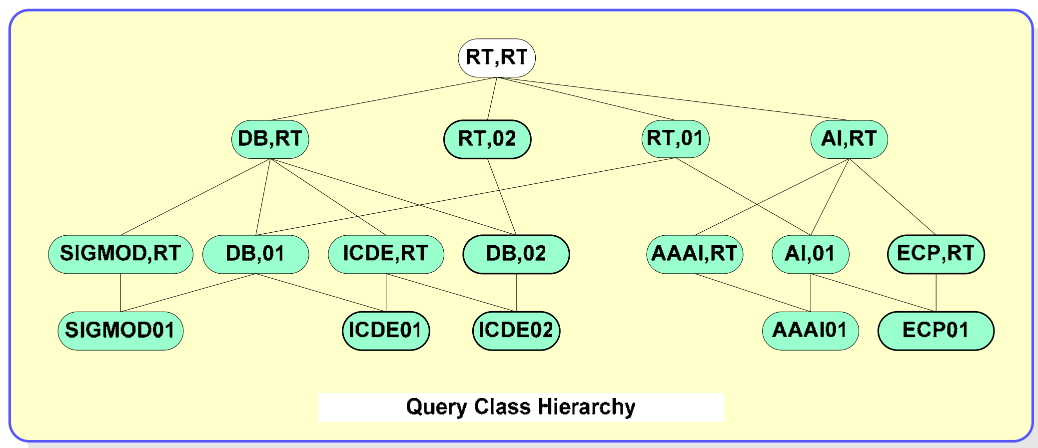
Attribute-Value Hierarchy:
An AV Hierarchy is a classification of the values of a particular attribute of the mediator relation. Leaf nodes in the hierarchy correspond to concrete values bound in a query.



AV Hierarchy for the Year A

AV Hierarchy for the Conference Attribute

Query Class: queries are grouped into classes by computing cartesian products over the AV Hierarchies.
A query class is a set of queries that all share a set of assignments of particular attributes to specific values.



Query Class Hierarchy

StatMiner

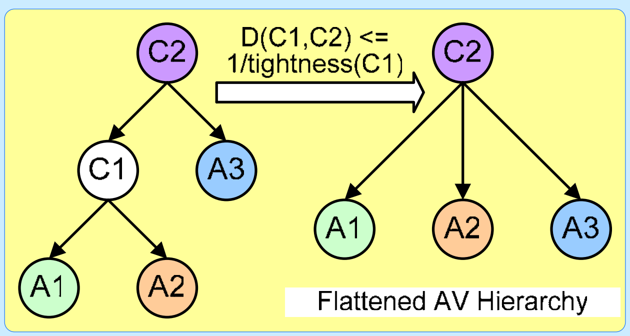
Learning AV Hierarchies

- Attribute values are extracted from the query list.
- Clustering similar attribute values leads to finding similar selection queries based on the similarity of their answer distributions over the sources.

$$d(Q1, Q2) = \sqrt{\sum_i [P(\hat{S}_i \mid Q1) - P(\hat{S}_i \mid Q2)]^2}$$

- The AV Hierarchies are generated using an agglomerative hierarchical clustering algorithm.
- They are then flattened according to their tightness.

$$tightness(C) = \frac{1}{\sum_{Q \in C} \frac{P(Q)}{P(C)} d(Q, C)}$$



Discovering Frequent Query Classes

- Candidate frequent query classes are identified using the anti-monotone property.
- Classes which are infrequently mapped are then removed.

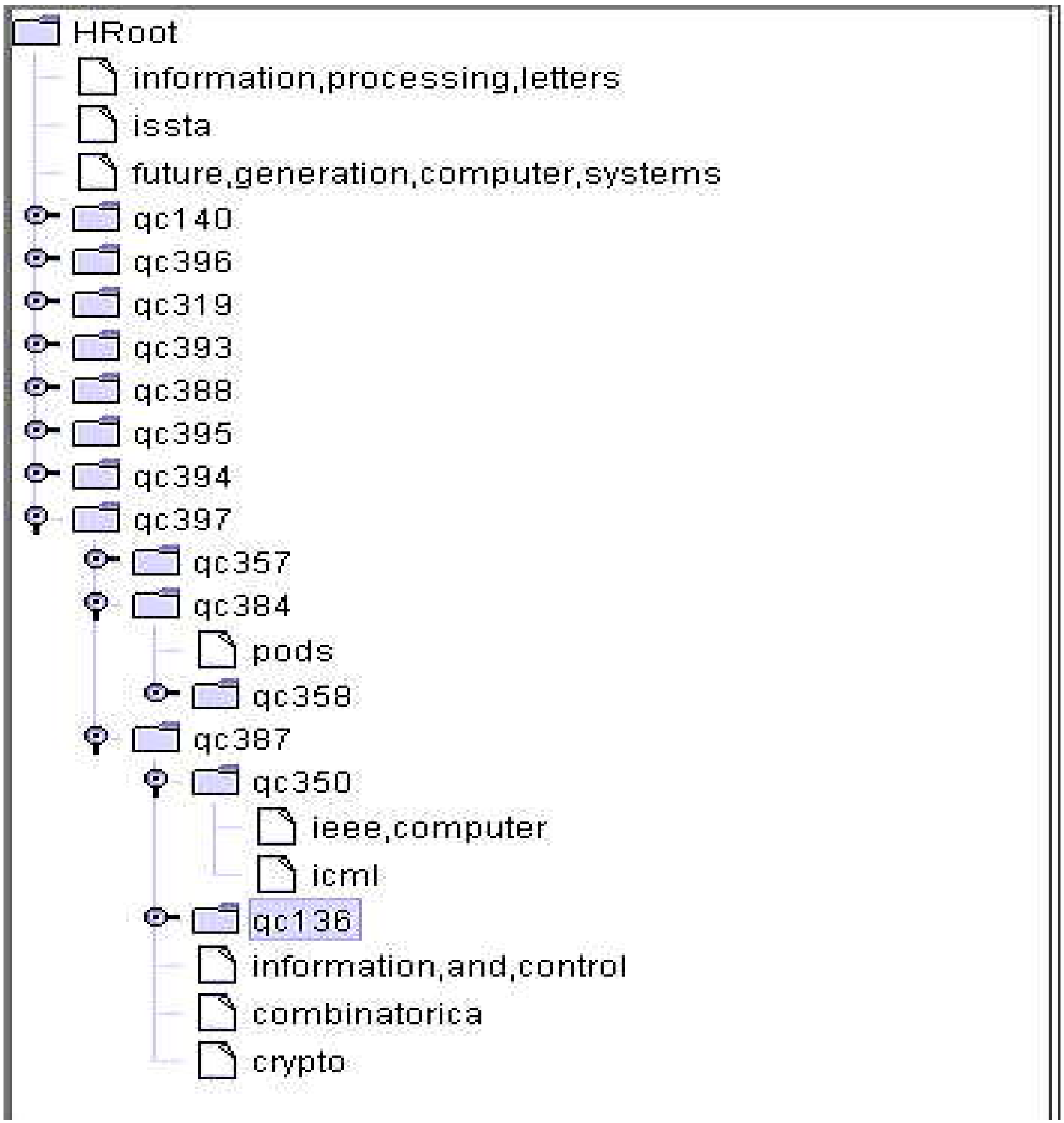
Learning Coverage and Overlap

Coverage and overlap statistics are computed for each frequent query class using a modified Apriori algorithm.

Raw Stats

$$P(\hat{S} \mid C) = \frac{\sum_{Q \in C} P(\hat{S} \mid Q) P(Q)}{P(C)}$$

Learned Conference Hierarchy



Using Coverage and Overlap Statistics to Rank Sources

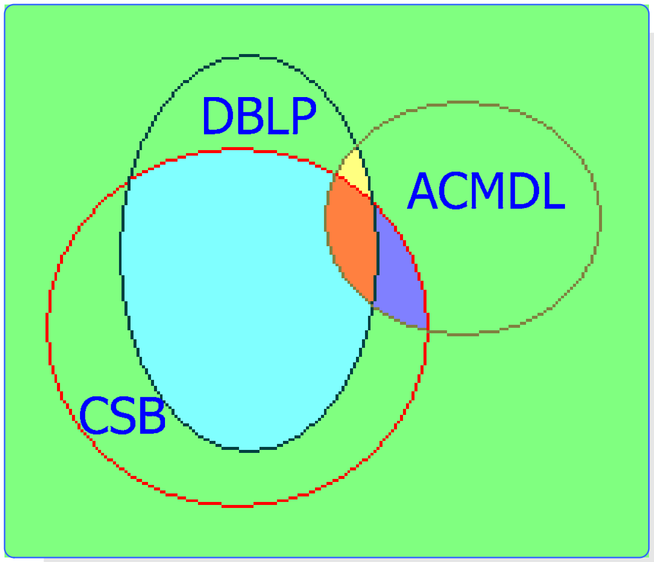
$$P(DBLP \mid Q, CSB) = P(DBLP \mid Q) - P(CSB \wedge DBLP \mid Q)$$

1. A new user query is mapped to a set of least general query classes.

2. The mediator estimates the statistics for the query using a weighted sum of the statistics of the mapped classes.

3. Data sources are ranked and called in order of relevance using the estimated statistics.
   In particular:
   - The most relevant source has highest coverage
   - The next best source has highest *residual* coverage

As a result, the maximum number of tuples are obtained while the least number of sources are called.



Example:
Here, CSB has highest coverage, followed by DBLP. However, since ACMDL has higher residual coverage than DBLP, the top 2 sources that would be called are CSB and ACMDL.

Latency statistics
(Or what good is coverage without good response time?)

- Sources vary significantly in terms of their response times
  - The response time depends both on the source itself, as well as the query that is asked of it
    - Specifically, what fields are bound in the selection query can make a difference
- ..So, learn statistics w.r.t. binding patterns

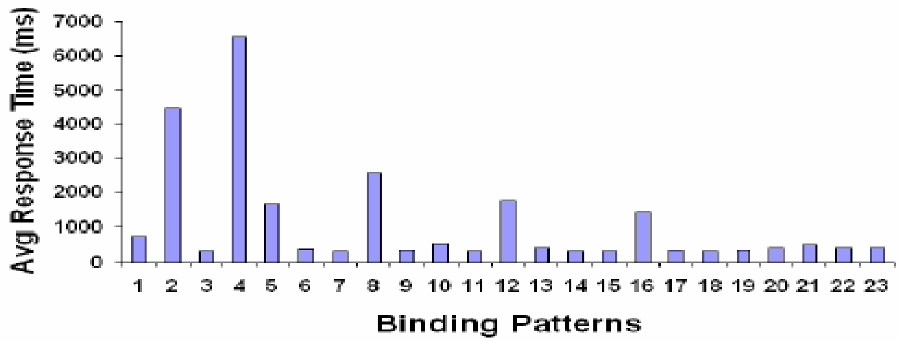


Figure 3: The average latency of the source DBLP shows significant variance between different binding patterns.

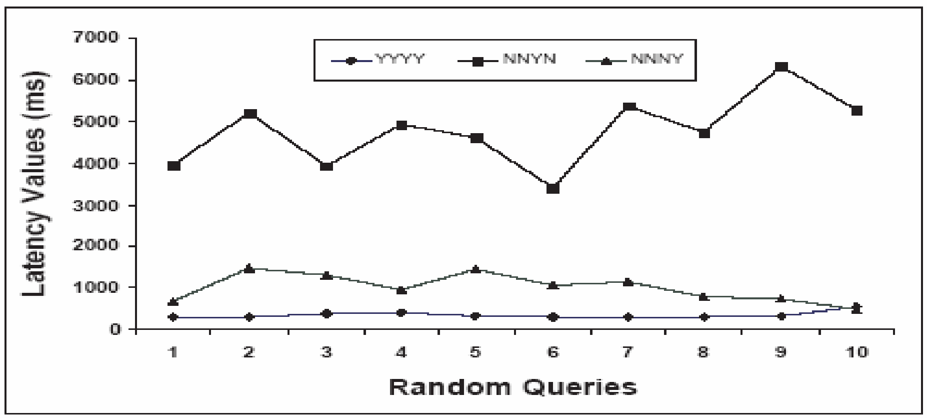


Figure 4: The Latency values of the source DBLP with 10 randomly selected queries for each of the 3 different binding patterns. The queries that fall in same binding patterns have similar latency values.

Query Binding Patterns

- A binding pattern refers to which arguments of a relational query are "bound"
  - Given a relation S(X,Y,Z)
    - A query S("Rao", Y, "Tom") has binding pattern *bfb*
    - A query S(X,Y, "TOM") has binding pattern *ffb*
- Binding patterns can be generalized to take "types" of bindings
  - E.g. S(X,Y,1) may be *ffn* (n being numeric binding) and
  - S(X,Y, "TOM") may be *ffs* (s being string binding)
- Sources tend to have different latencies based on the binding pattern
  - In extreme cases, certain binding patterns may have infinite latency (i.e., you *are not allowed to ask that query)*
    - Called "infeasible" binding patterns

(Digression)

- LCWs are the “qualitative” versions of quantitative coverage/overlap statistics
- Feasible binding patterns are “qualitative” versions of quantitative latency statistics

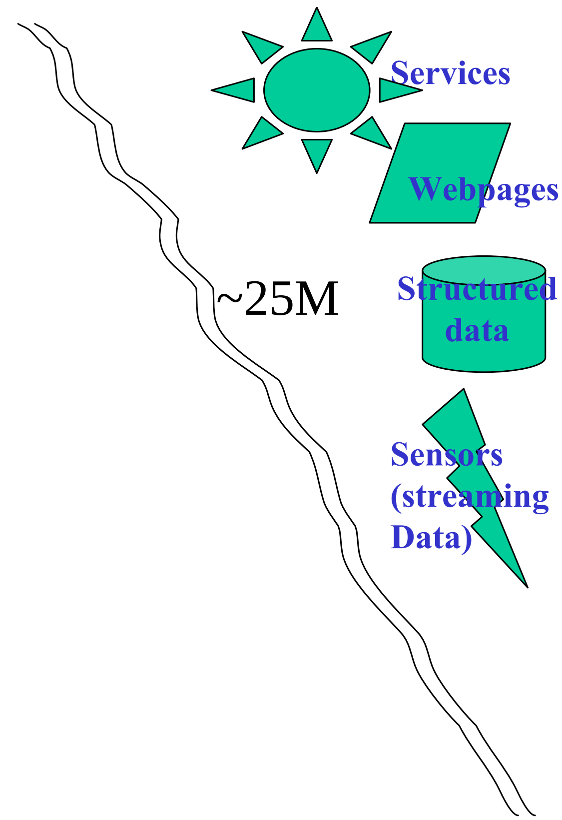
Combining coverage and response time

- Qn: How do we define an optimal plan in the context of both coverage/overlap and response time requirements?
  - An instance of "multi-objective" optimization
    - General solution involves presenting a set of "pareto-optimal" solutions to the user and let her decide
      - Pareto-optimal set is a set of solutions where no solution is dominated by another one in *all optimization dimensions* (i.e., both better coverage and lower response time)
    - Another idea is to combine both objectives into a single weighted objective

$$Util(S_i) = ResidualCoverage(S_i|Q) \times \gamma^{Latency(S_i|Q)}$$

$$\begin{aligned} Util(S_i) &= \omega \times \log(ResidualCoverage(S_i|Q)) + \\ &\quad (1-\omega) \times (-Latency(S_i|Q)) \end{aligned}$$

Query Processing Challenges



--Warehouse Model--
--Get all the data and put into a local DB
--Support structured queries on the warehouse

Supporting Imprecision/ Incompleteness/Uncertainty

--Mediator Model--
--Design a mediator
--Reformulate queries
--Access sources
--Collate results

Query reformulation
Optimizing Access to Sources

Indexing with Structure

--Search Model--
--Materialize the pages
--crawl &index them
--include them in search results

Kambhampati & Knoblock