Big Data Integration

Dr. Salahuddin Shaikh

What is "Big Data Integration?"

- Big data integration = Big data + data integration

- Data integration: easy access to multiple data sources [DHI12]
  - Virtual: mediated schema, query redirection, link + fuse answers
  - Warehouse: materialized data, easy querying, consistency issues

- Big data: all about the V's ☺
  - Size: large **volume** of data, collected and analyzed at high **velocity**
  - Complexity: huge **variety** of data, of questionable **veracity**

What is "Big Data Integration?"

- Big data integration = Big data + data integration

- Data integration: easy access to multiple data sources [DHI12]
  - Virtual: mediated schema, query redirection, link + fuse answers
  - Warehouse: materialized data, easy querying, consistency issues

- Big data in the context of data integration: still about the V's ☺
  - Size: large **volume** of sources, changing at high **velocity**
  - Complexity: huge **variety** of sources, of questionable **veracity**

Why Do We Need "Big Data Integration?"

- Building web-scale knowledge bases

Google knowledge graph

ProBase

MSR knowledge base

A Little Knowledge Goes a Long Way.

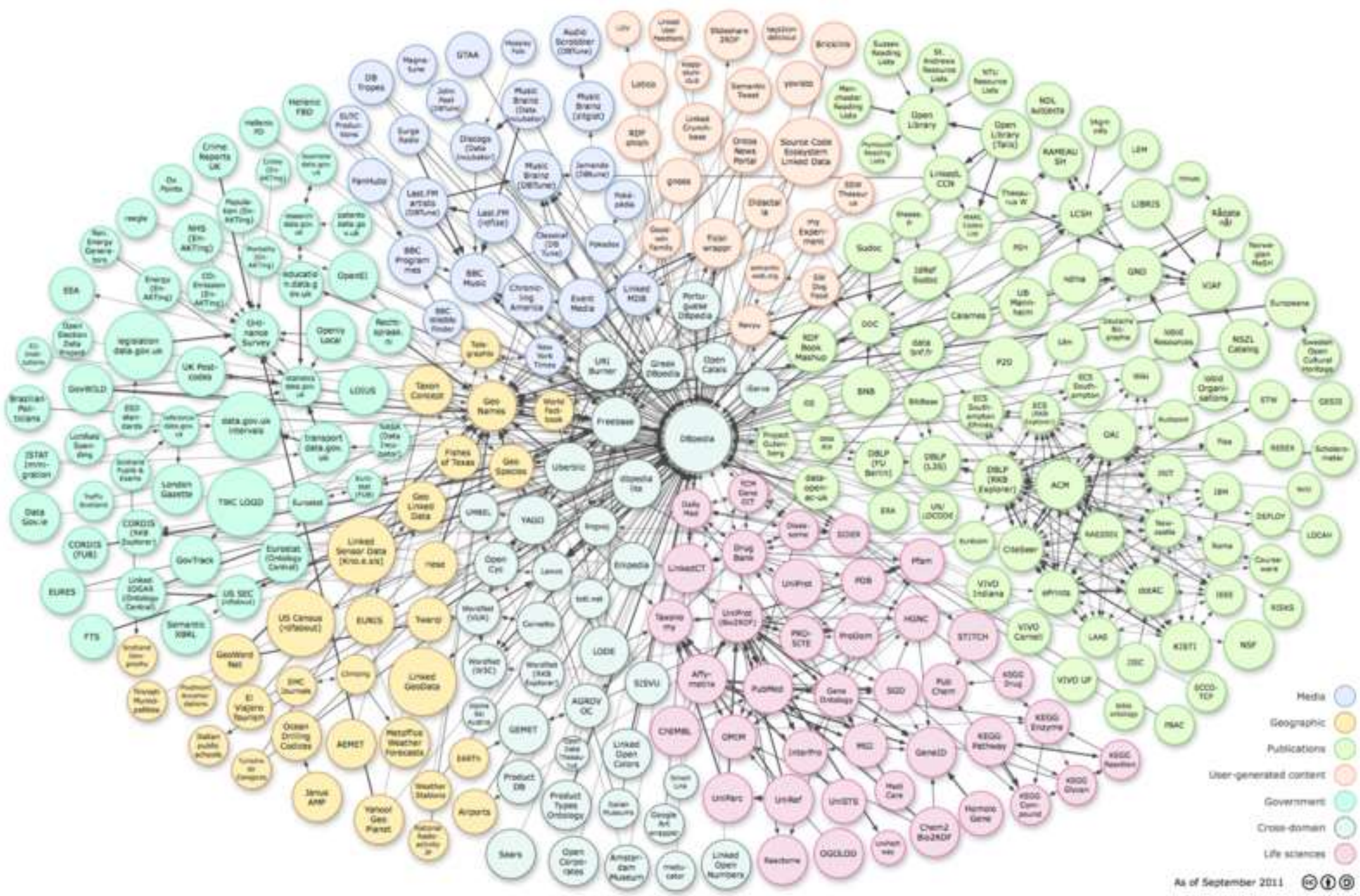
Freebase

Domain	ID	Topics	Facts
Music	/music	24M	161M
Media	/media_common	7M	23M
Books	/book	6M	37M
People	/people	3M	13M

yago
select knowledge

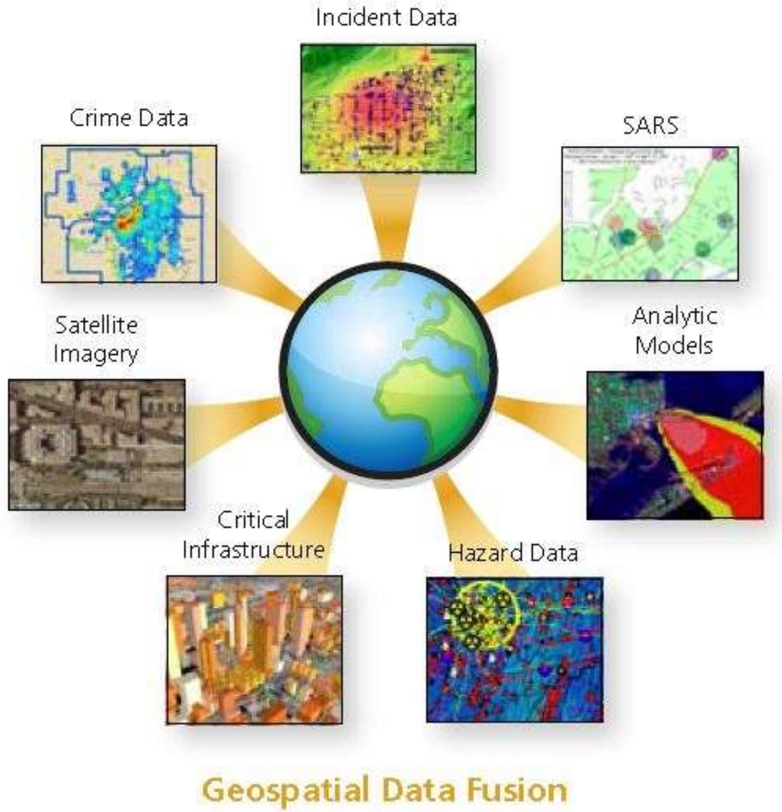
Why Do We Need "Big Data Integration?"

- Reasoning over linked data



Why Do We Need "Big Data Integration?"

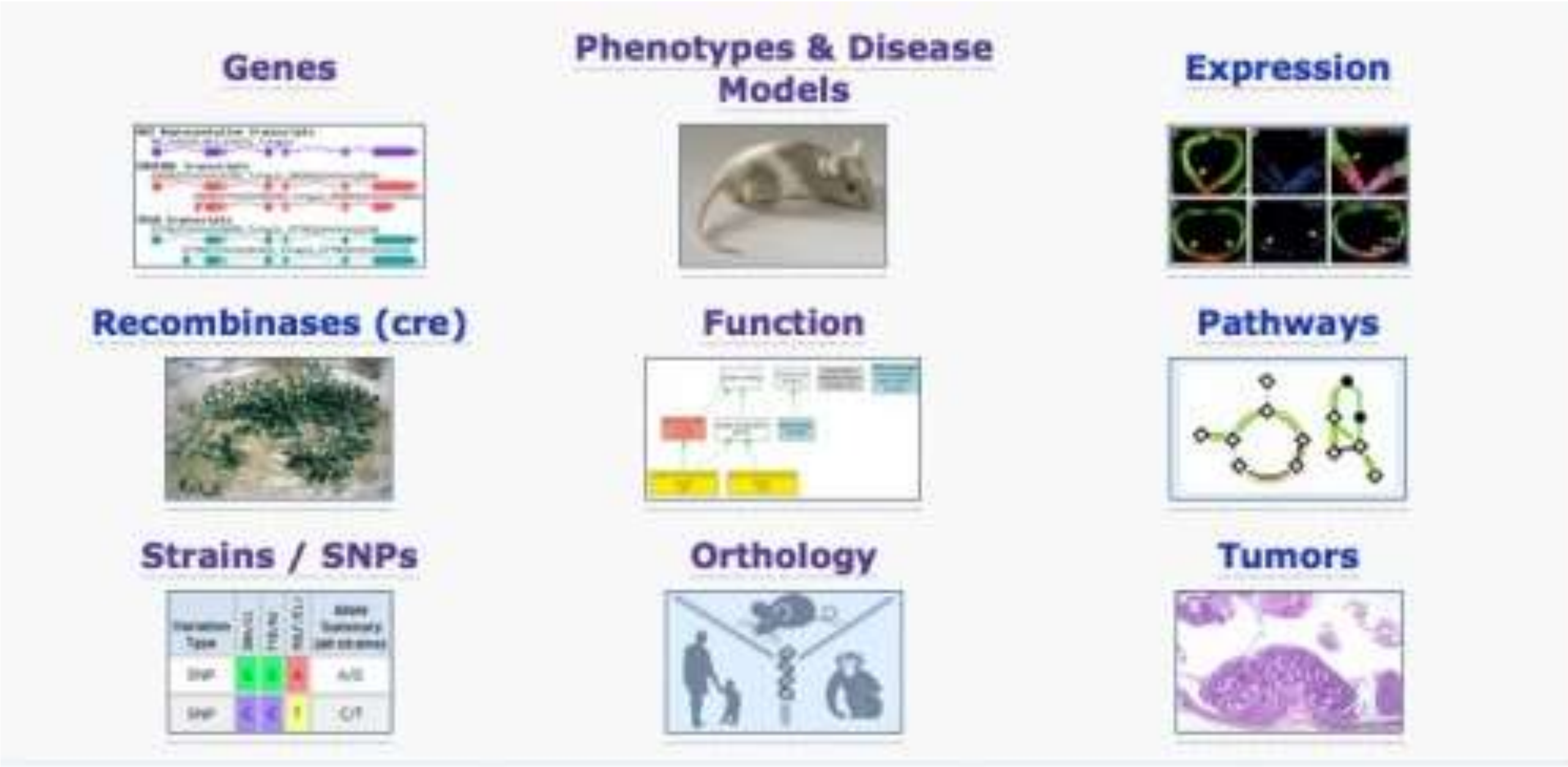
- Geo-spatial data fusion



http://axiomamuse.wordpress.com/2011/04/18/

Why Do We Need "Big Data Integration?"

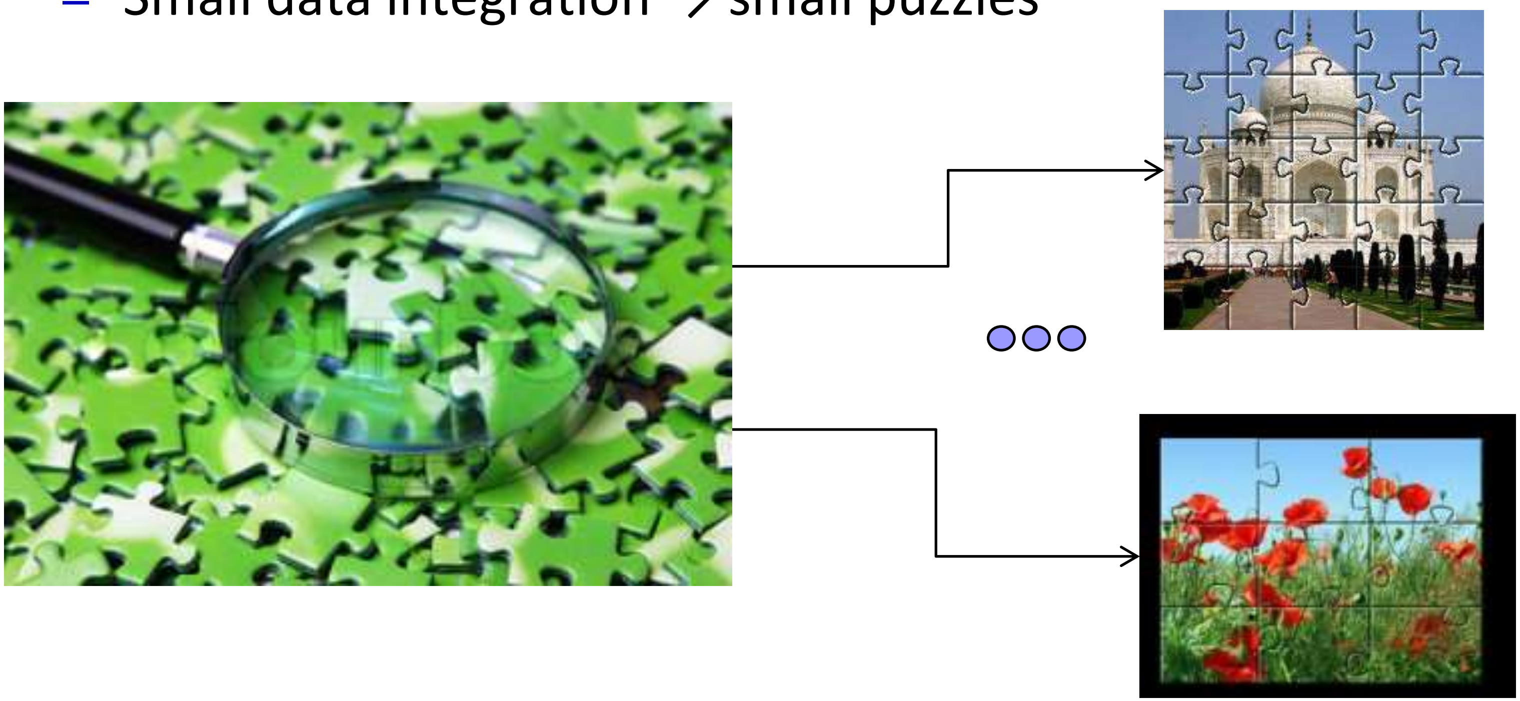
- Scientific data analysis



http://scienceline.org/2012/01/from-index-cards-to-information-overload/

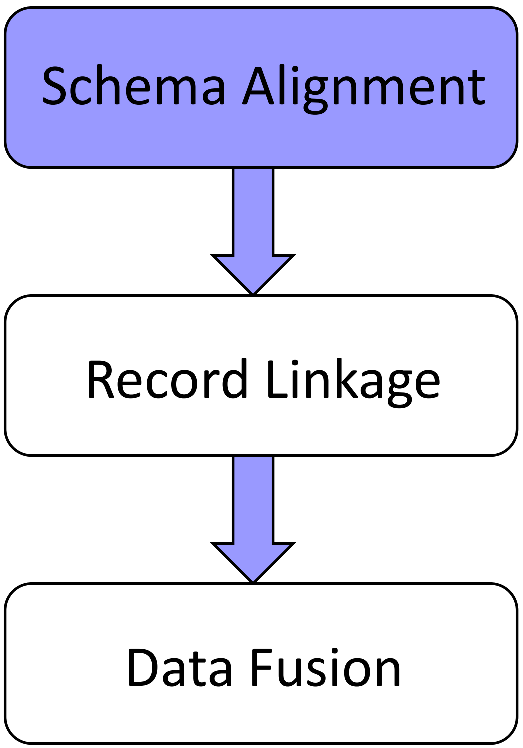
“Small” Data Integration: Why is it Hard?

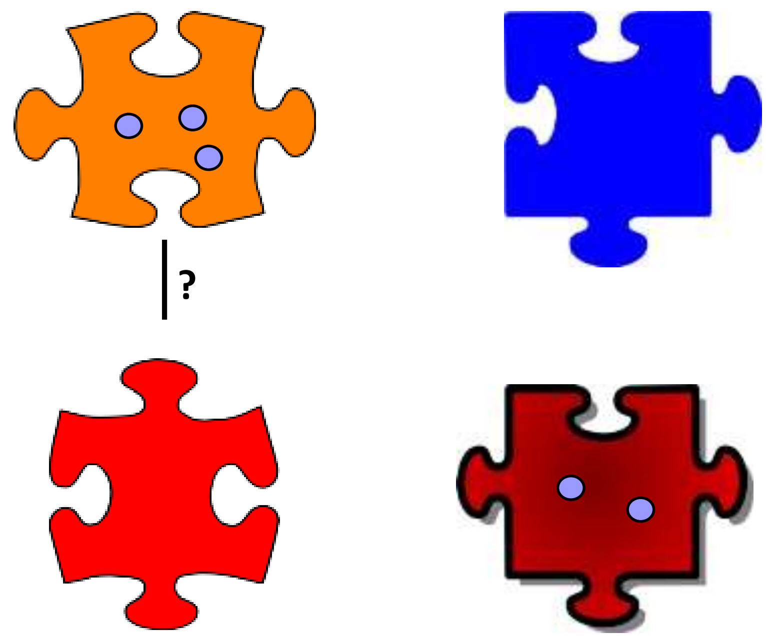
- Data integration = solving lots of jigsaw puzzles
  - Each jigsaw puzzle (e.g., Taj Mahal) is an **integrated entity**
  - Each type of puzzle (e.g., flowers) is an **entity domain**
  - Small data integration → small puzzles



“Small” Data Integration: How is it Done?

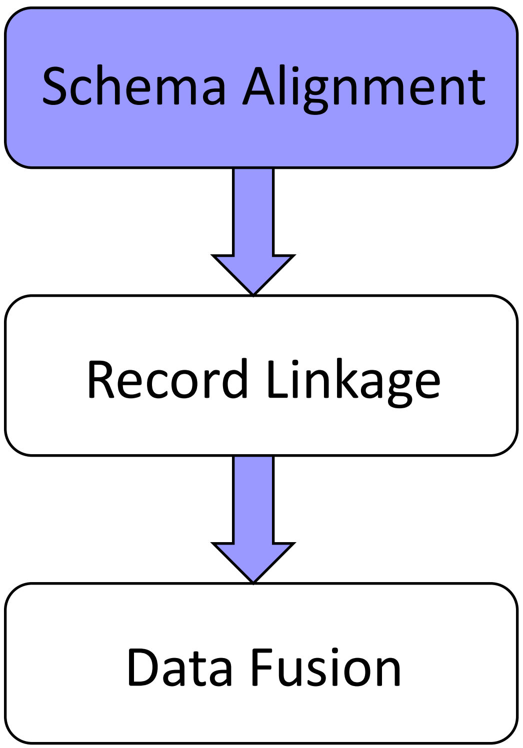
- “Small” data integration: alignment + linkage + fusion
  - Schema alignment: mapping of **structure** (e.g., shape)

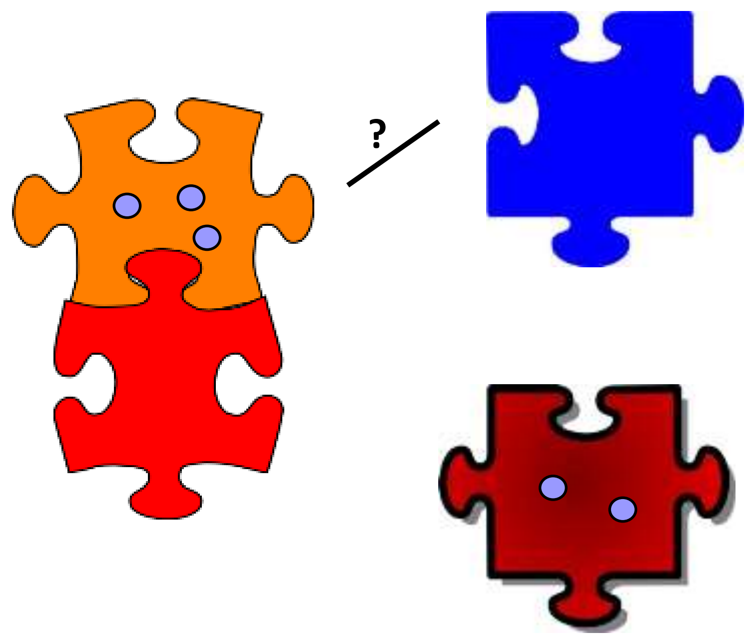




“Small” Data Integration: How is it Done?

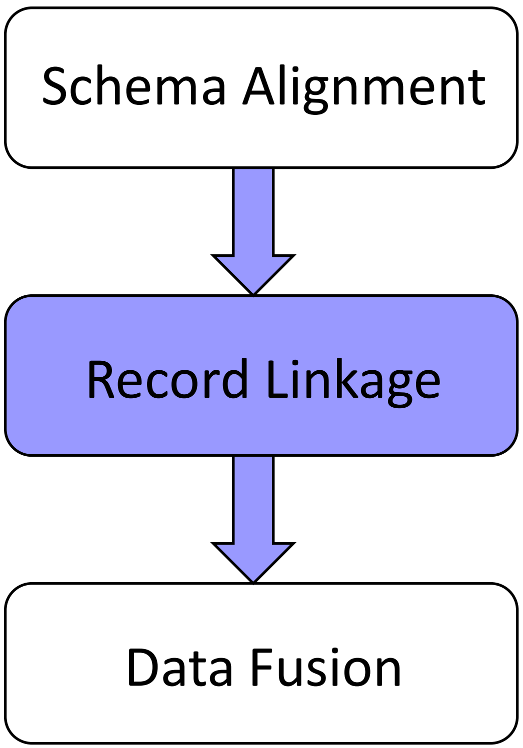
- “Small” data integration: alignment + linkage + fusion
  - Schema alignment: mapping of **structure** (e.g., shape)

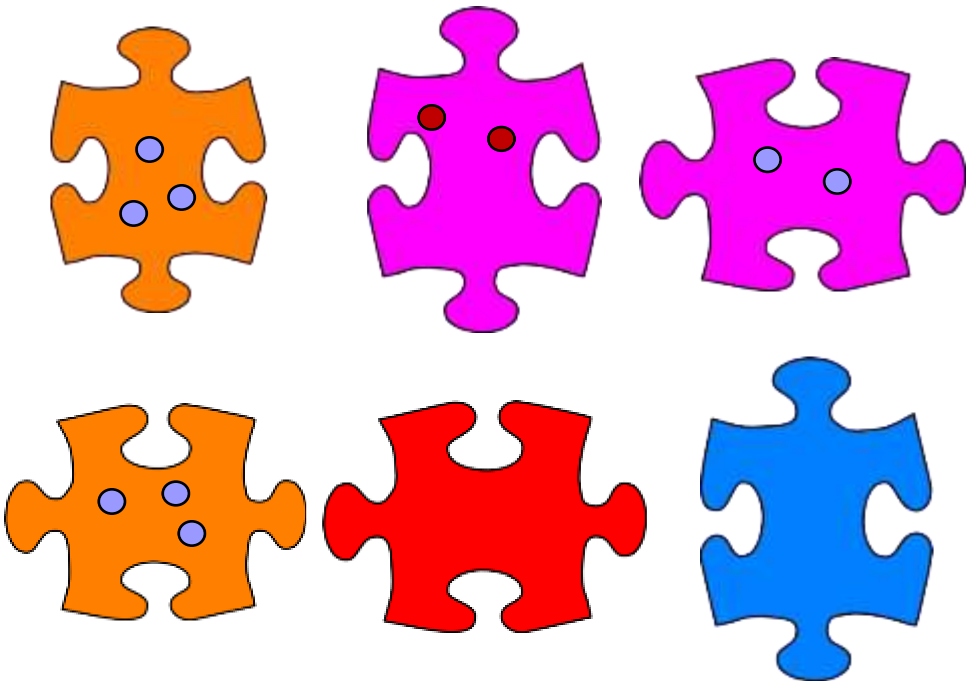




“Small” Data Integration: How is it Done?

- “Small” data integration: alignment + linkage + fusion
  - Record linkage: matching based on **identifying content** (e.g., color)

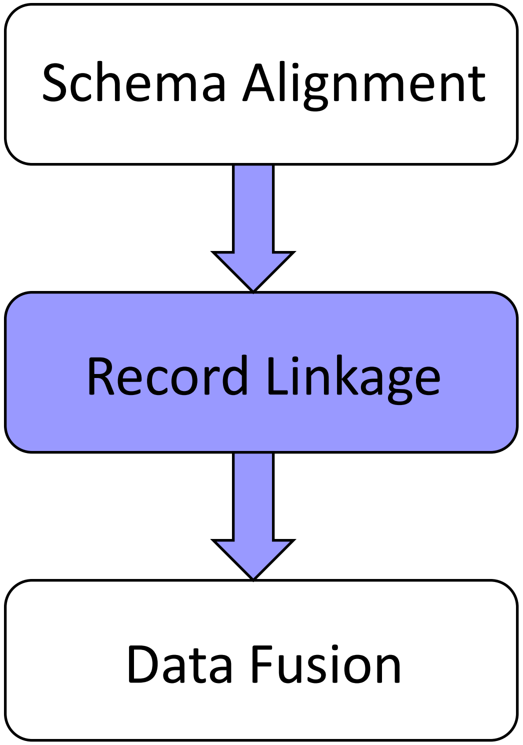


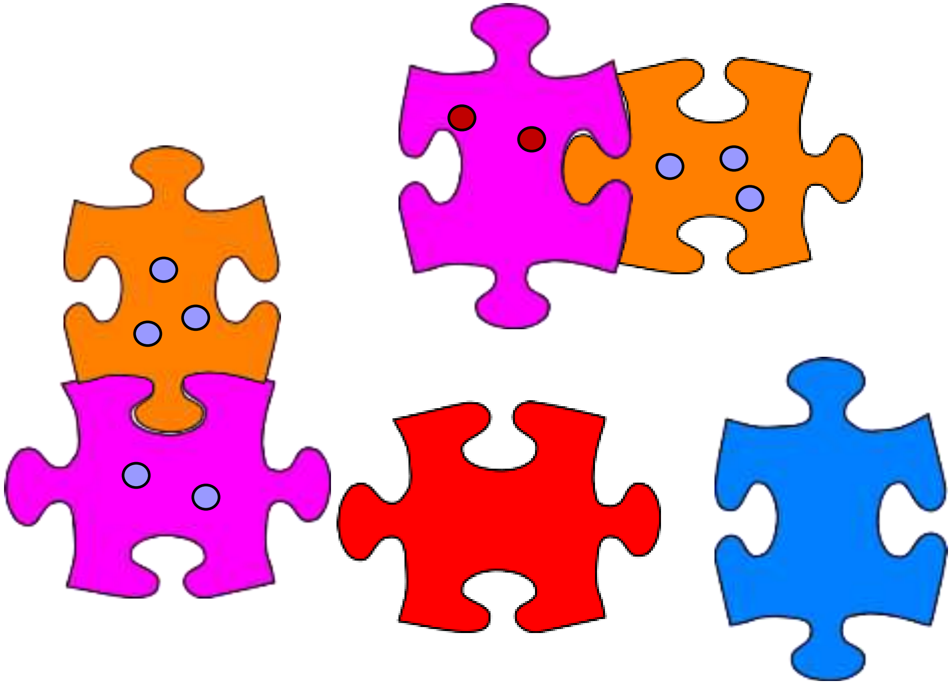


X

"Small" Data Integration: How is it Done?

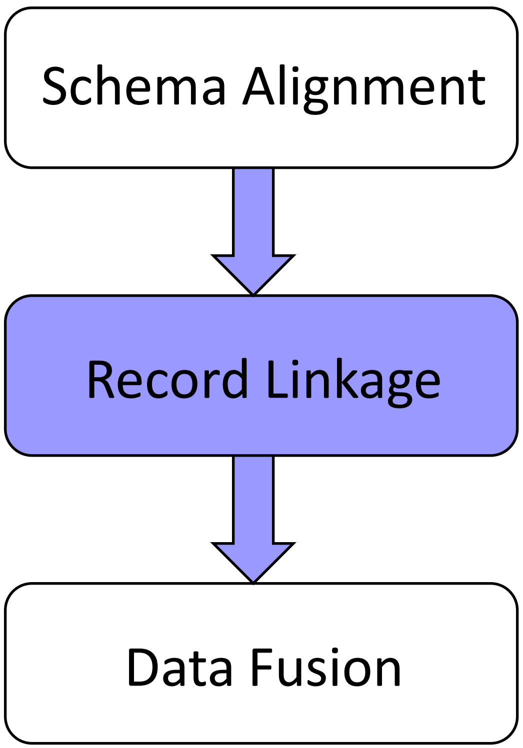
- "Small" data integration: alignment + linkage + fusion
  - Record linkage: matching based on **identifying content** (e.g., color)

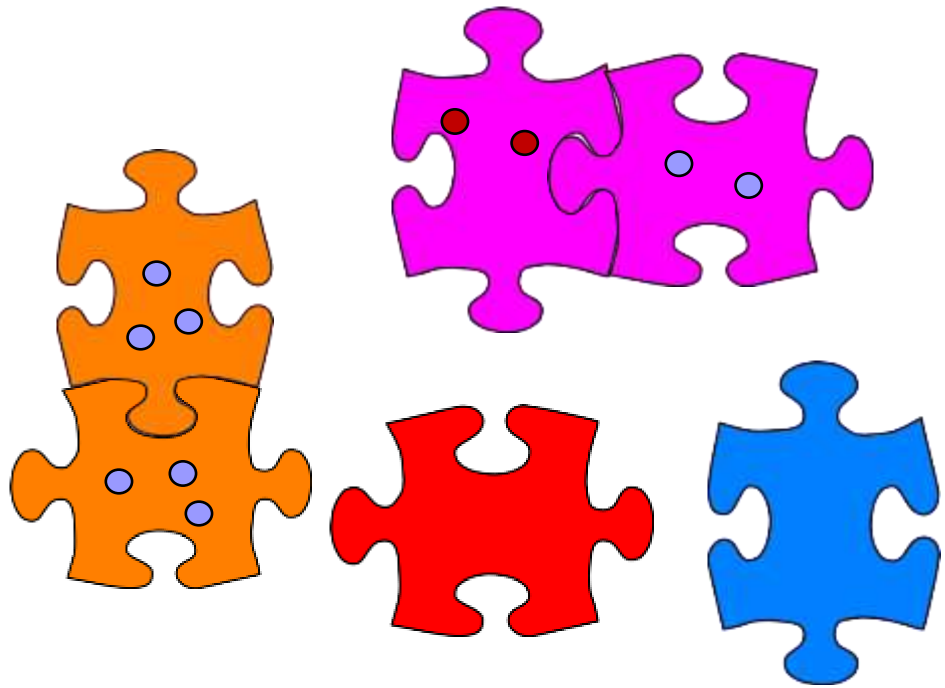




“Small” Data Integration: How is it Done?

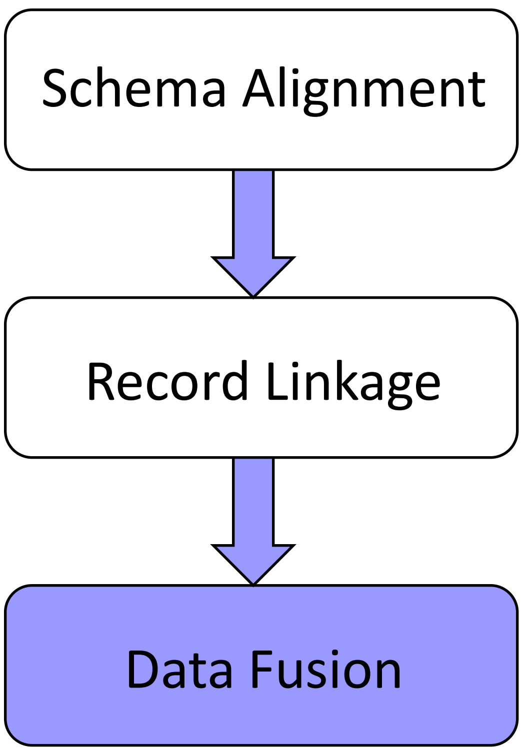
- “Small” data integration: alignment + linkage + fusion
  - Record linkage: matching based on **identifying content** (e.g., color)

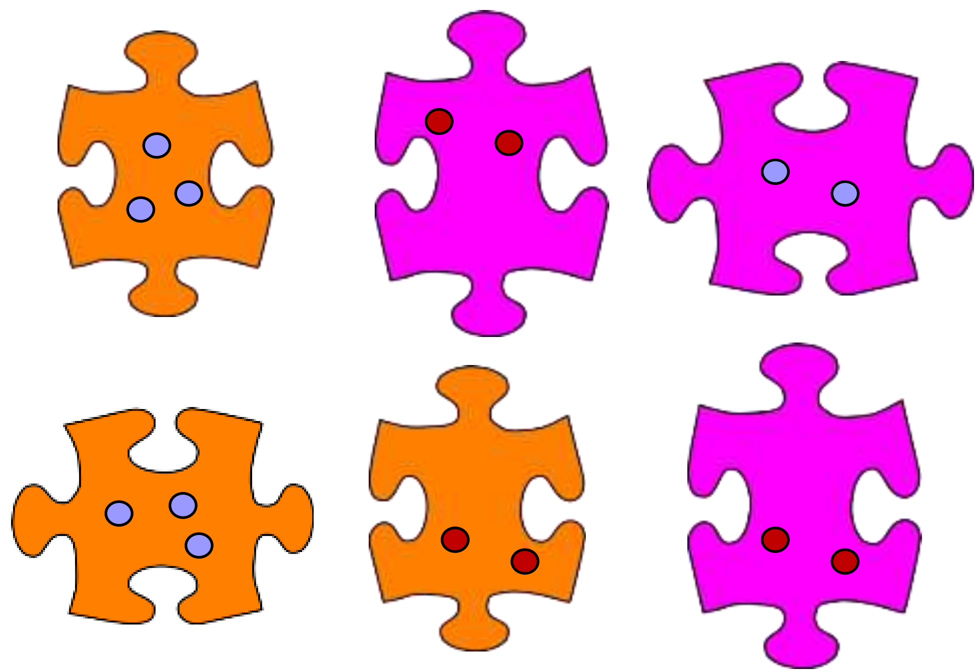




"Small" Data Integration: How is it Done?

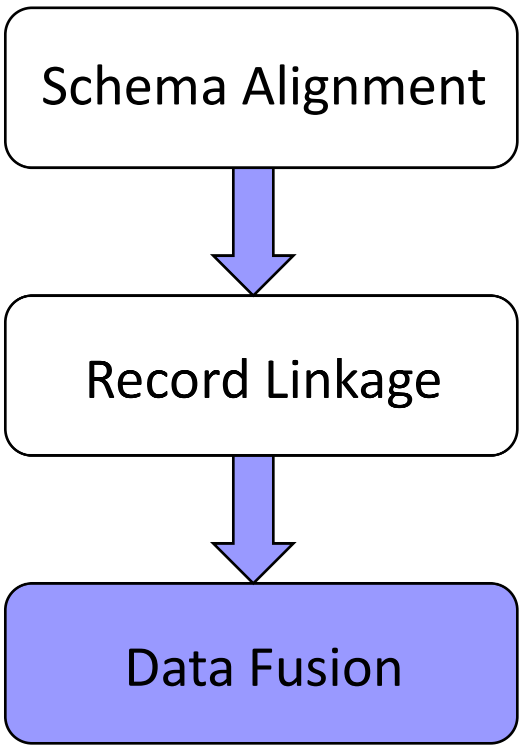
- "Small" data integration: alignment + linkage + fusion
  - Data fusion: reconciliation of **non-identifying content** (e.g., dots)

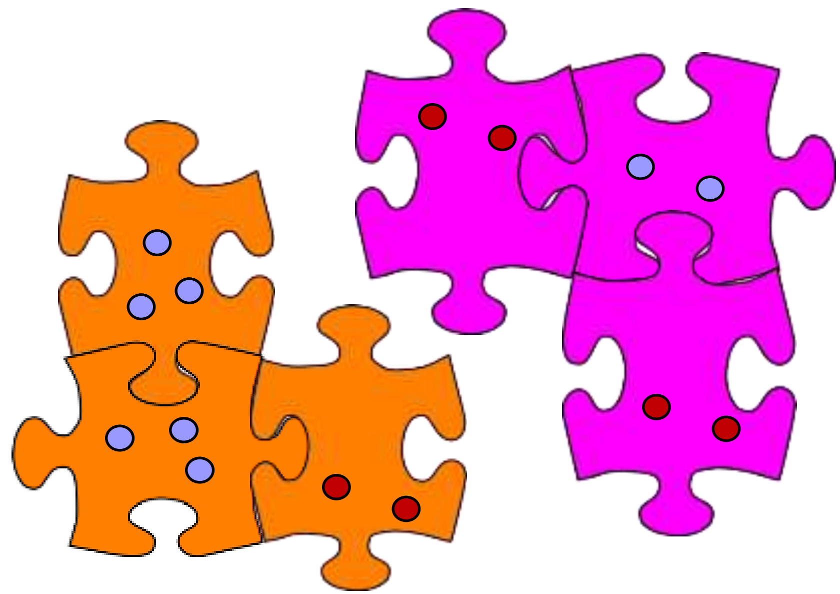




"Small" Data Integration: How is it Done?

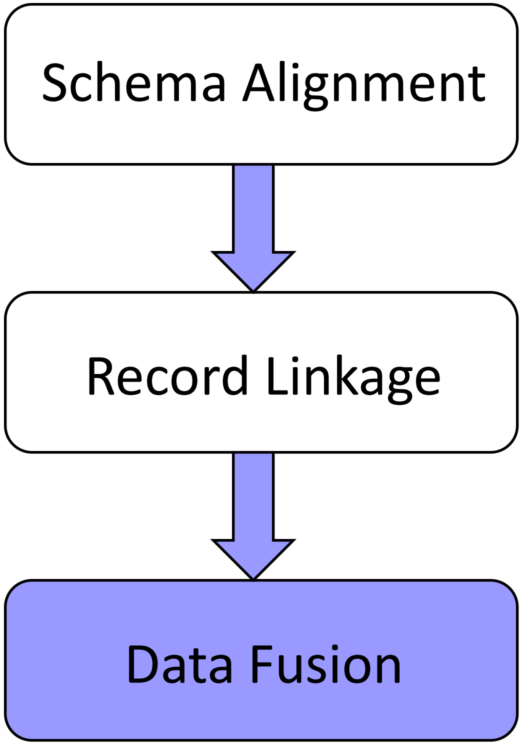
- "Small" data integration: alignment + linkage + fusion
  - Data fusion: reconciliation of **non-identifying content** (e.g., dots)

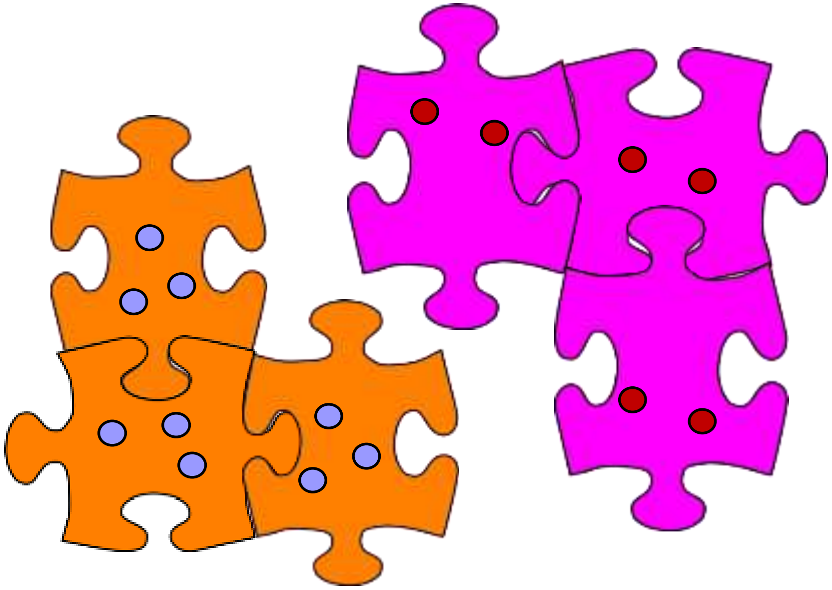




“Small” Data Integration: How is it Done?

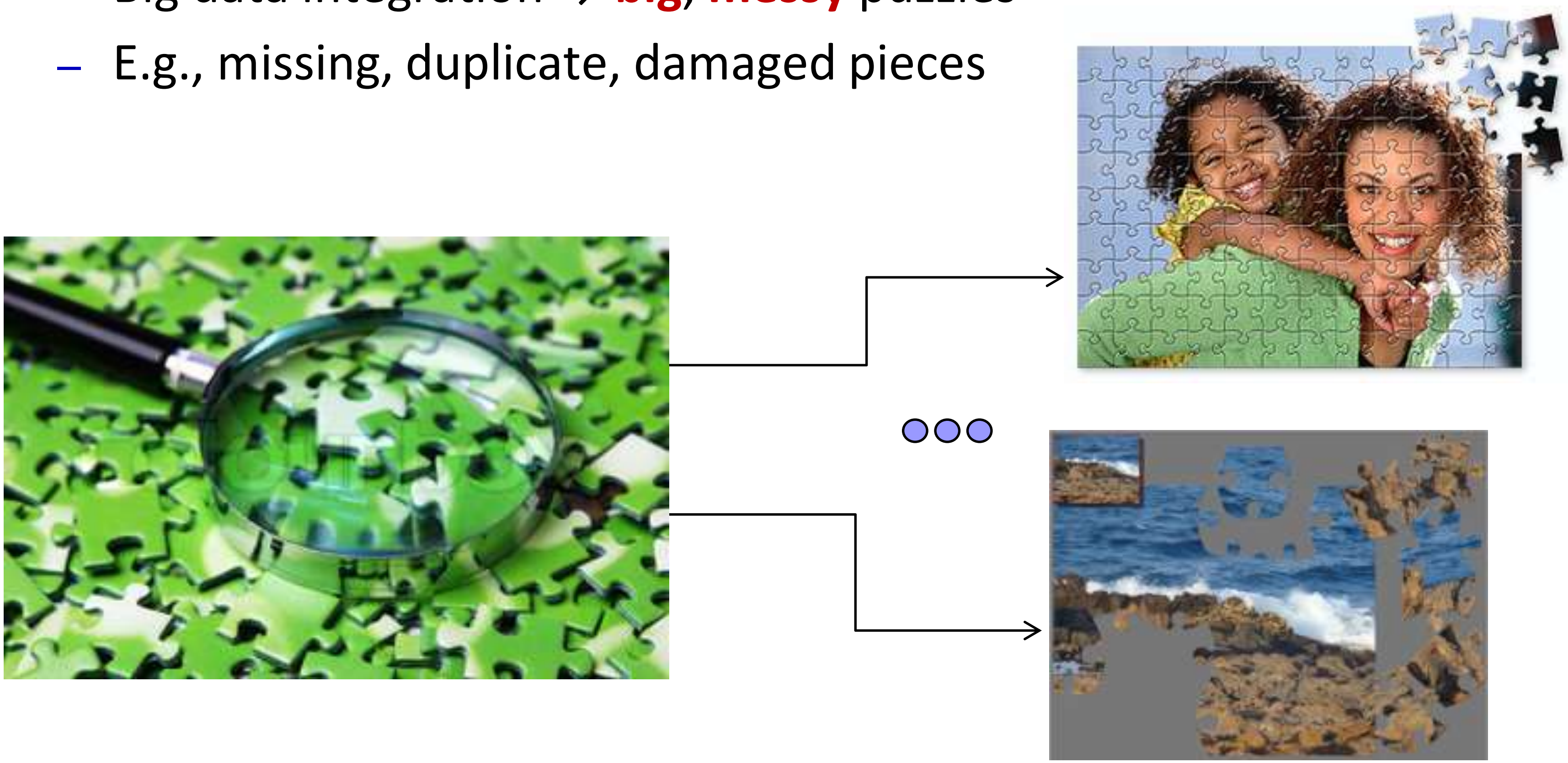
- “Small” data integration: alignment + linkage + fusion
  - Data fusion: reconciliation of **non-identifying content** (e.g., dots)





BDI: Why is it Challenging?

- Data integration = solving lots of jigsaw puzzles
  - Big data integration → **big**, **messy** puzzles
  - E.g., missing, duplicate, damaged pieces



BDI: Why is it Challenging?

- Number of structured sources: **Volume**
  - 154 million high quality relational tables on the web [CHW+08]
  - 10s of millions of high quality deep web sources [MKK+08]
  - 10s of millions of useful relational tables from web lists [EMH09]

- Challenges:
  - Difficult to do schema alignment
  - Expensive to warehouse all the integrated data
  - Infeasible to support virtual integration

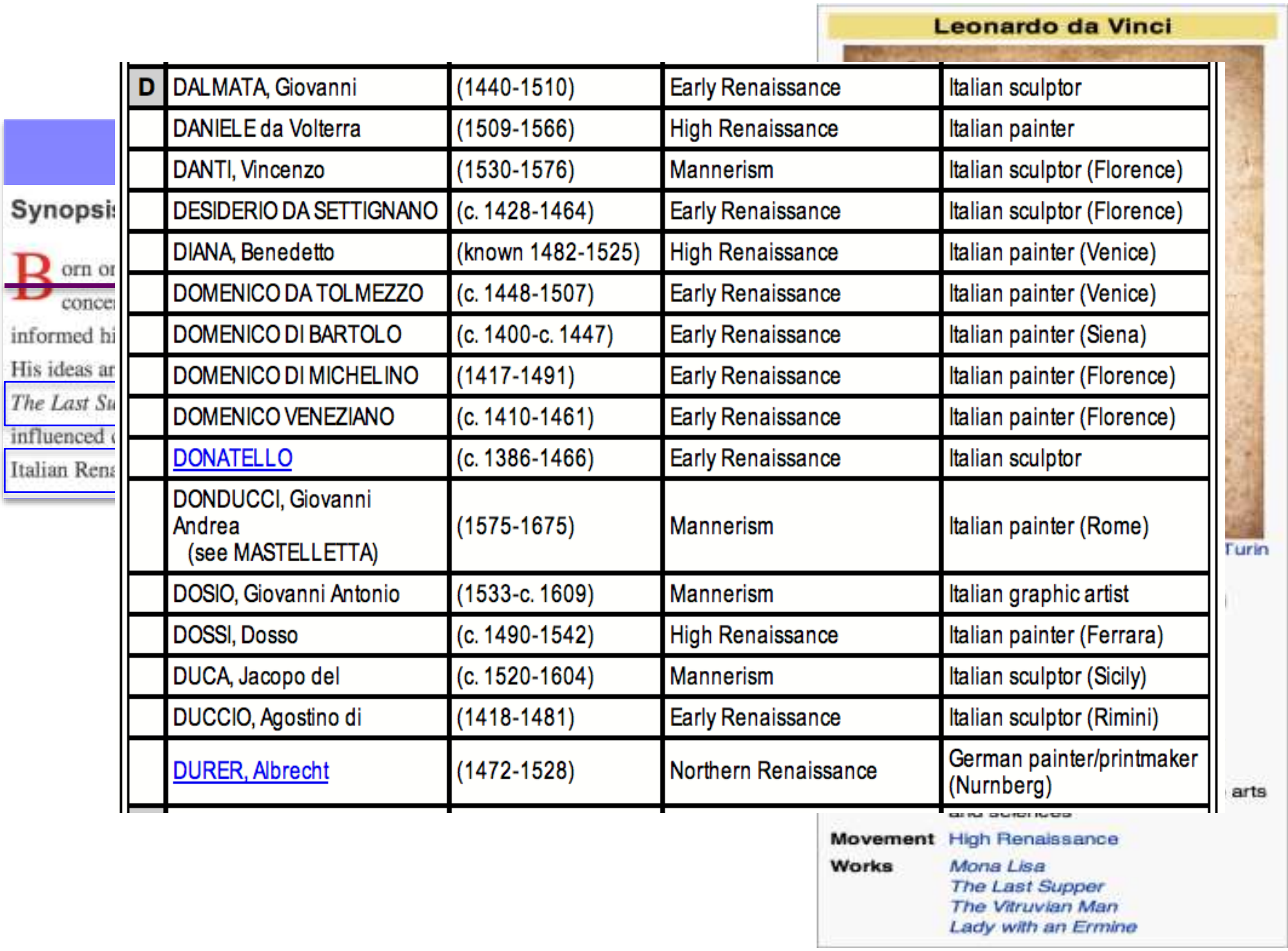
BDI: Why is it Challenging?

- Rate of change in structured sources: **Velocity**
  - 43,000 – 96,000 deep web sources (with HTML forms) [B01]
  - 450,000 databases, 1.25M query interfaces on the web [CHZ05]
  - 10s of millions of high quality deep web sources [MKK+08]
  - Many sources provide rapidly changing data, e.g., stock prices

- Challenges:
  - Difficult to understand evolution of semantics
  - Extremely expensive to warehouse data history
  - Infeasible to capture rapid data changes in a timely fashion

BDI: Why is it Challenging?

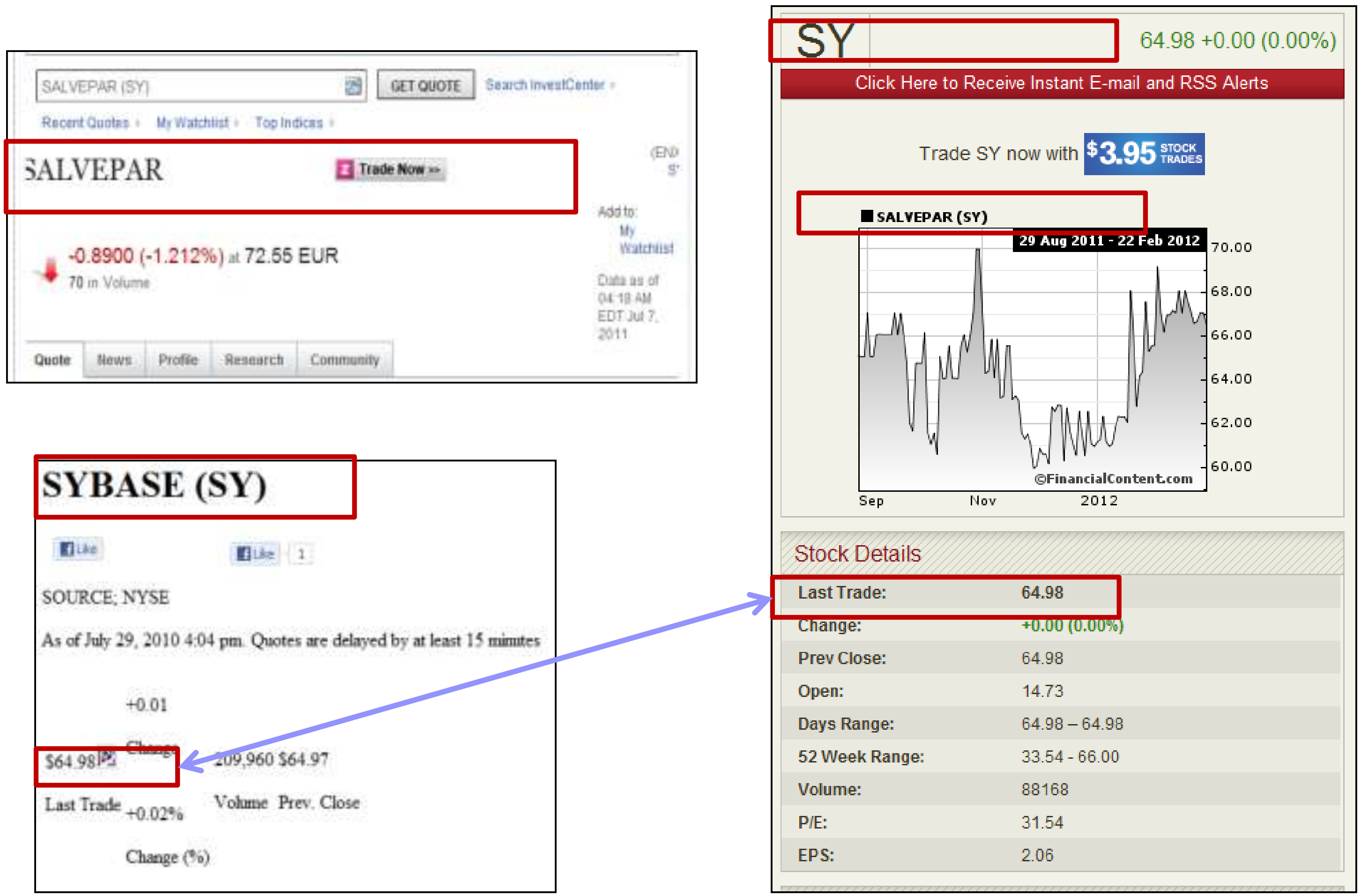
- Representation differences among sources: **Variety**



D	DALMATA, Giovanni	(1440-1510)	Early Renaissance	Italian sculptor
	DANIELE da Volterra	(1509-1566)	High Renaissance	Italian painter
	DANTI, Vincenzo	(1530-1576)	Mannerism	Italian sculptor (Florence)
	DESIDERIO DA SETTIGNANO	(c. 1428-1464)	Early Renaissance	Italian sculptor (Florence)
	DIANA, Benedetto	(known 1482-1525)	High Renaissance	Italian painter (Venice)
	DOMENICO DA TOLMEZZO	(c. 1448-1507)	Early Renaissance	Italian painter (Venice)
	DOMENICO DI BARTOLO	(c. 1400-c. 1447)	Early Renaissance	Italian painter (Siena)
	DOMENICO DI MICHELINO	(1417-1491)	Early Renaissance	Italian painter (Florence)
	DOMENICO VENEZIANO	(c. 1410-1461)	Early Renaissance	Italian painter (Florence)
	DONATELLO	(c. 1386-1466)	Early Renaissance	Italian sculptor
	DONDUCCI, Giovanni Andrea (see MASTELLETTA)	(1575-1675)	Mannerism	Italian painter (Rome)
	DOSIO, Giovanni Antonio	(1533-c. 1609)	Mannerism	Italian graphic artist
	DOSSI, Dosso	(c. 1490-1542)	High Renaissance	Italian painter (Ferrara)
	DUCA, Jacopo del	(c. 1520-1604)	Mannerism	Italian sculptor (Sicily)
	DUCCIO, Agostino di	(1418-1481)	Early Renaissance	Italian sculptor (Rimini)
	DURER, Albrecht	(1472-1528)	Northern Renaissance	German painter/printmaker (Nurnberg)

BDI: Why is it Challenging?

- Poor data quality of deep web sources [LDL+13]: **Veracity**

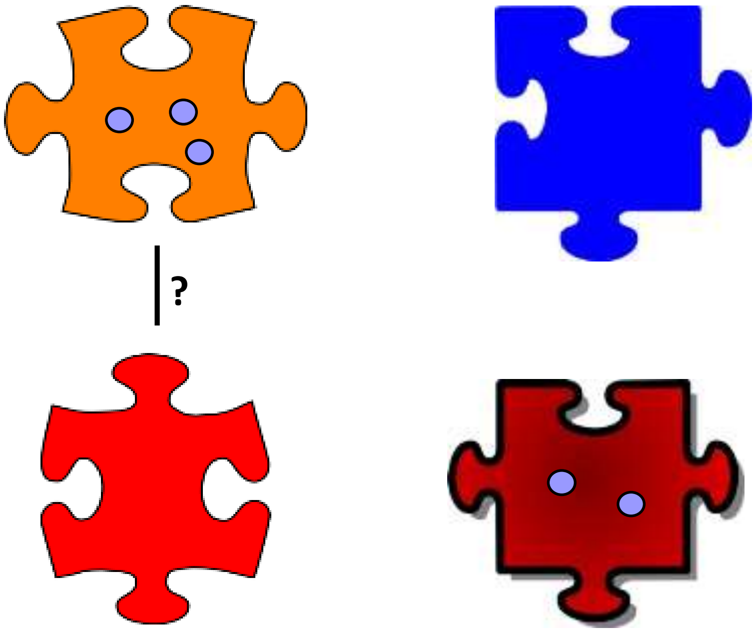


Outline

- Motivation
- Schema alignment
  - Overview
  - Techniques for big data
- Record linkage
- Data fusion

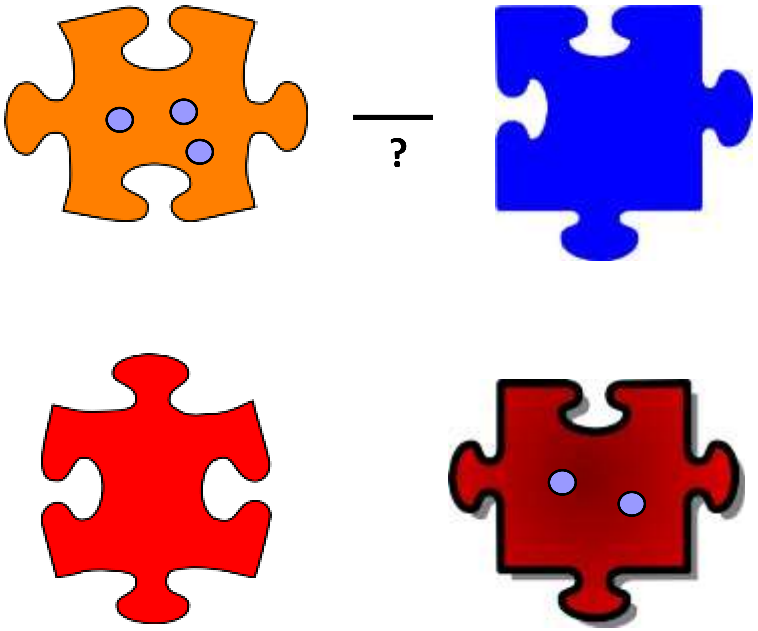
Schema Alignment

- Matching based on structure



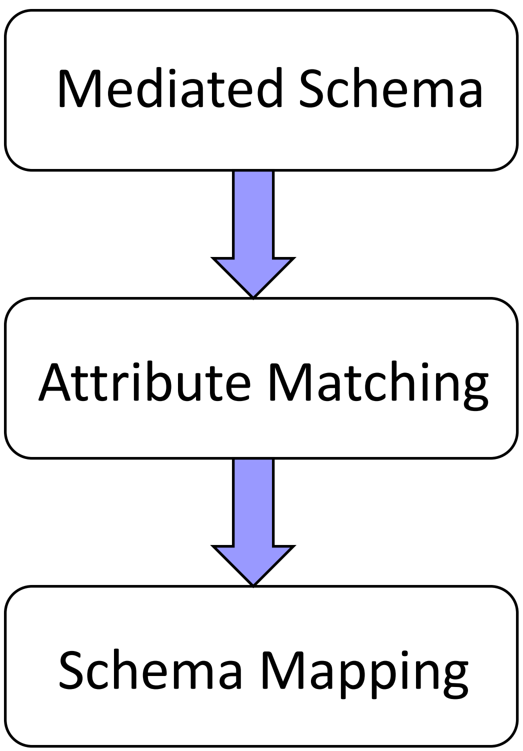
Schema Alignment

- Matching based on structure



Schema Alignment: Three Steps [BBR11]

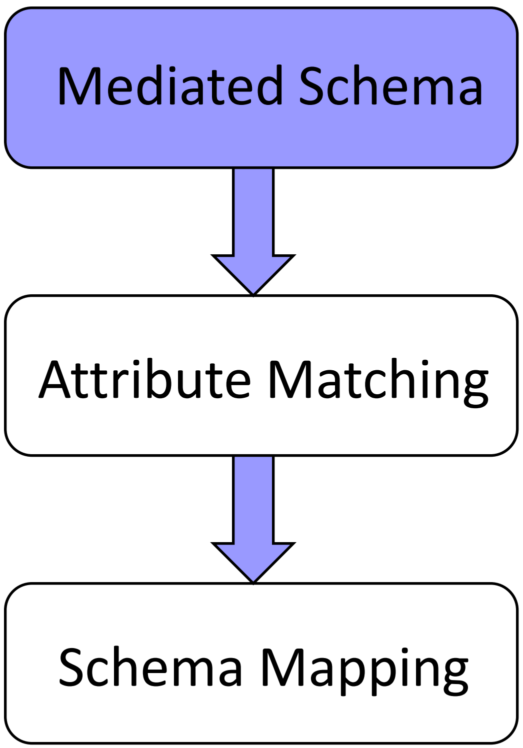
- Schema alignment: mediated schema + matching + mapping
  - Enables linkage, fusion to be semantically meaningful



S1	(name, games, runs)
S2	(name, team, score)
S3	a: (id, name); b: (id, team, runs)
S4	(name, club, matches)
S5	(name, team, matches)

Schema Alignment: Three Steps

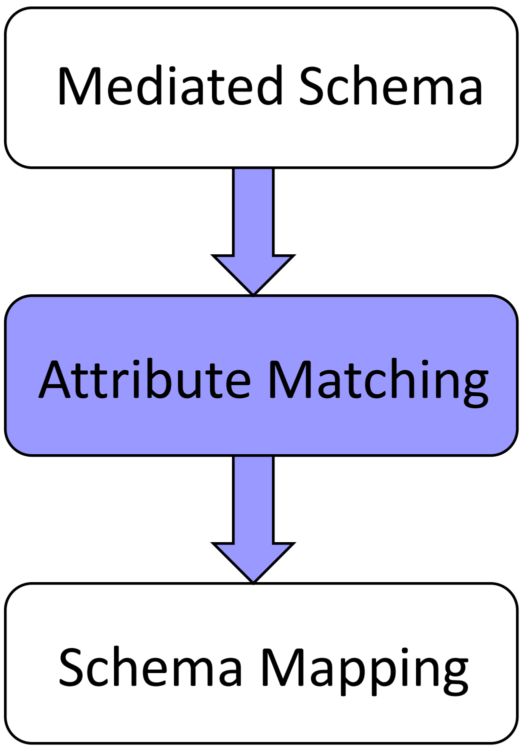
- Schema alignment: mediated schema + matching + mapping
  - Enables domain specific modeling



S1	(name, games, runs)
S2	(name, team, score)
S3	a: (id, name); b: (id, team, runs)
S4	(name, club, matches)
S5	(name, team, matches)
MS	**(n, t, g, s)**

Schema Alignment: Three Steps

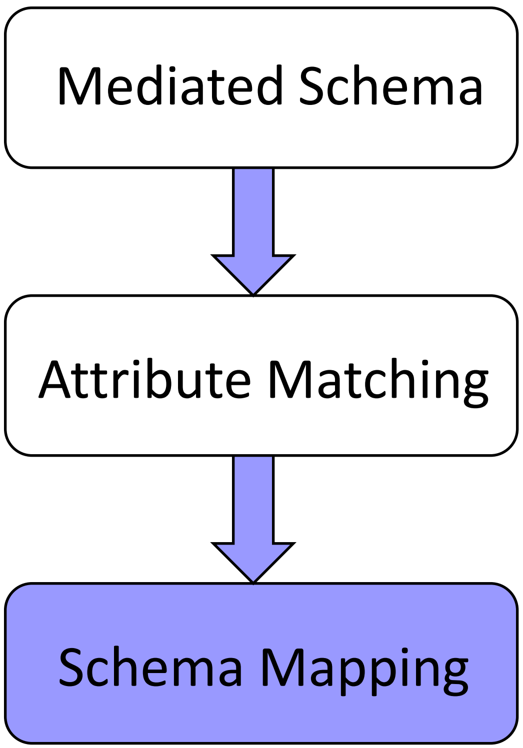
- Schema alignment: mediated schema + matching + mapping
  - Identifies correspondences between schema attributes



S1	(name, games, runs)
S2	(name, team, score)
S3	a: (id, name); b: (id, team, runs)
S4	(name, club, matches)
S5	(name, team, matches)
MS	**(n, t, g, s)**
MSAM	**MS.n**: S1.name, S2.name, … **MS.t**: S2.team, S4.club, … **MS.g**: S1.games, S4.matches, … **MS.s**: S1.runs, S2.score, …

Schema Alignment: Three Steps

- Schema alignment: mediated schema + matching + mapping
  - Specifies transformation between records in different schemas



S1	(name, games, runs)
S2	(name, team, score)
S3	a: (id, name); b: (id, team, runs)
S4	(name, club, matches)
S5	(name, team, matches)
MS	**(n, t, g, s)**
MSSM	∀n, t, g, s (**MS(n, t, g, s)** → S1(n, g, s) | S2(n, t, s) | ∃ i (S3a(i, n) & S3b(i, t, s)) | S4(n, t, g) | S5(n, t, g))

Outline

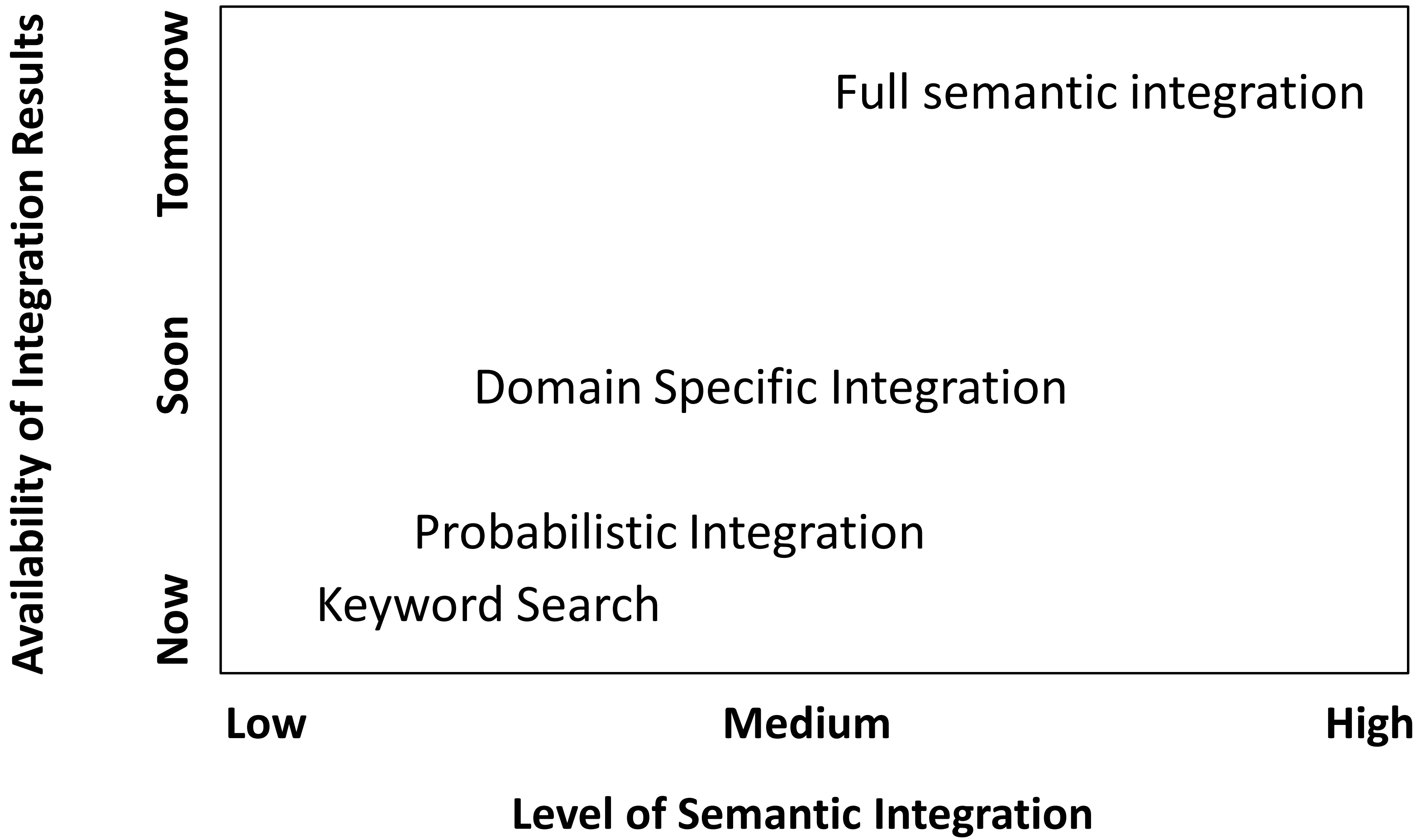
- Motivation
- Schema alignment
  - Overview
  - Techniques for big data
- Record linkage
- Data fusion

BDI: Schema Alignment

- **Volume, Variety**
  - Integrating deep web query interfaces [WYD+04, CHZ05]
  - Dataspace systems [FHM05, HFM06, DHY07]
  - Keyword search based data integration [TJM+08]
  - Crawl, index deep web data [MKK+08]
  - Extract structured data from web tables [CHW+08, PS12, DFG+12] and web lists [GS09, EMH09]

- **Velocity**
  - Keyword search-based dynamic data integration [TIP10]

Space of Strategies



WebTables [CHW+08]

- Background: Google crawl of the surface web, reported in 2008
  - 154M good relational tables, 5.4M attribute names, 2.6M schemas
- ACSDb
  - (schema, count)

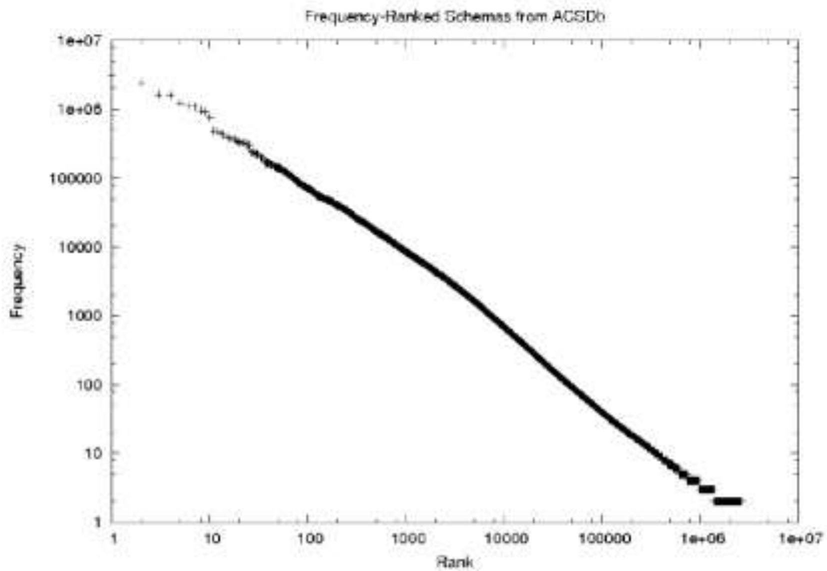


Figure 3: Distribution of frequency-ordered unique schemas in the ACSDb, with rank-order on the x-axis, and schema frequency on the y-axis. Both rank and frequency axes have a log scale.

WebTables: Keyword Ranking [CHW+08]

- Goal: Rank tables on web in response to query keywords
  - Not web pages, not individual records

- Challenges:
  - Web page features apply ambiguously to embedded tables
  - Web tables on a page may not all be relevant to a query
  - Web tables have specific features (e.g., schema elements)

WebTables: Keyword Ranking

- FeatureRank: use table specific features
  - Query independent features
  - Query dependent features
  - Linear regression estimator
  - Heavily weighted features

rows
cols
has-header?
# of NULLs in table
document-search rank of source page
hits on header
hits on leftmost column
hits on second-to-leftmost column
hits on table body

- Result quality: fraction of high scoring relevant tables

k	Naïve	FeatureRank
10	0.26	0.43
20	0.33	0.56
30	0.34	0.66

WebTables: Keyword Ranking

- SchemaRank: also include **schema coherency**
  - Use point-wise mutual information (pmi) derived from ACSDb
  - p(S) = fraction of unique schemas containing attributes S
  - pmi(a,b) = log(p(a,b)/(p(a)*p(b)))
  - Coherency = average pmi(a,b) over all a, b in attrs(R)

- Result quality: fraction of high scoring relevant tables

k	Naïve	FeatureRank	SchemaRank
10	0.26	0.43	0.47
20	0.33	0.56	0.59
30	0.34	0.66	0.68

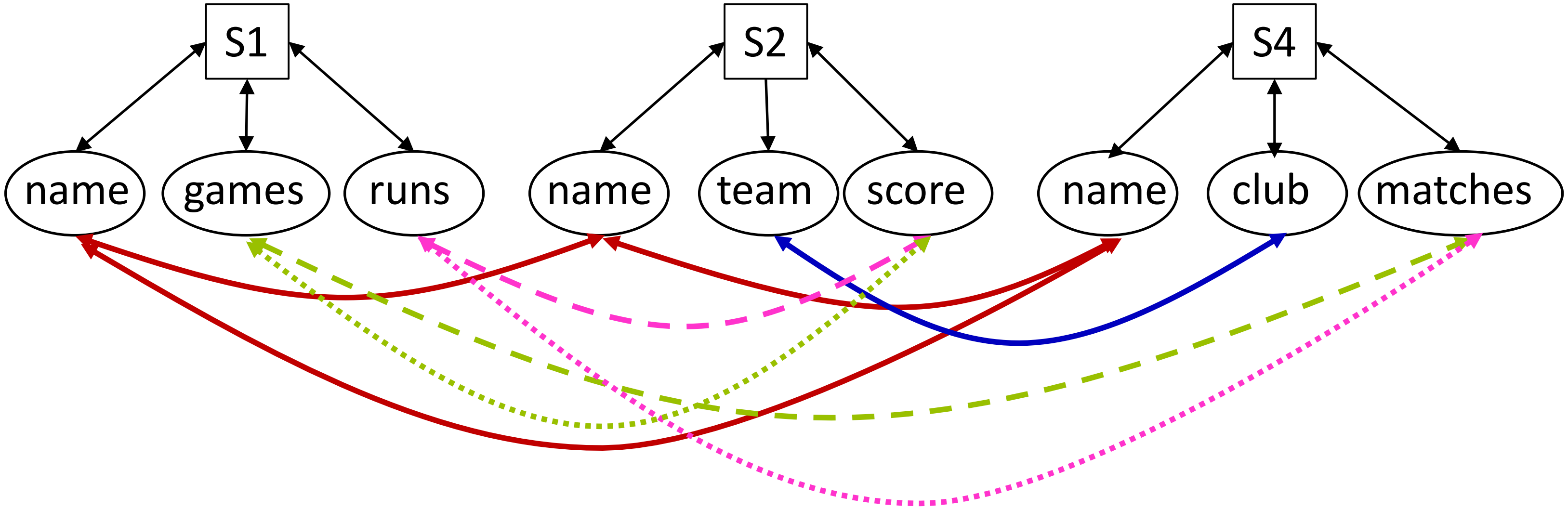
Dataspace Approach [FHM05, HFM06]

- Motivation: SDI approach (as-is) is infeasible for BDI
  - **Volume**, **variety** of sources → unacceptable up-front modeling cost
  - **Velocity** of sources → expensive to maintain integration results

- Key insight: **pay-as-you-go** approach may be feasible
  - Start with simple, universally useful service
  - Iteratively add complexity when and where needed [JFH08]

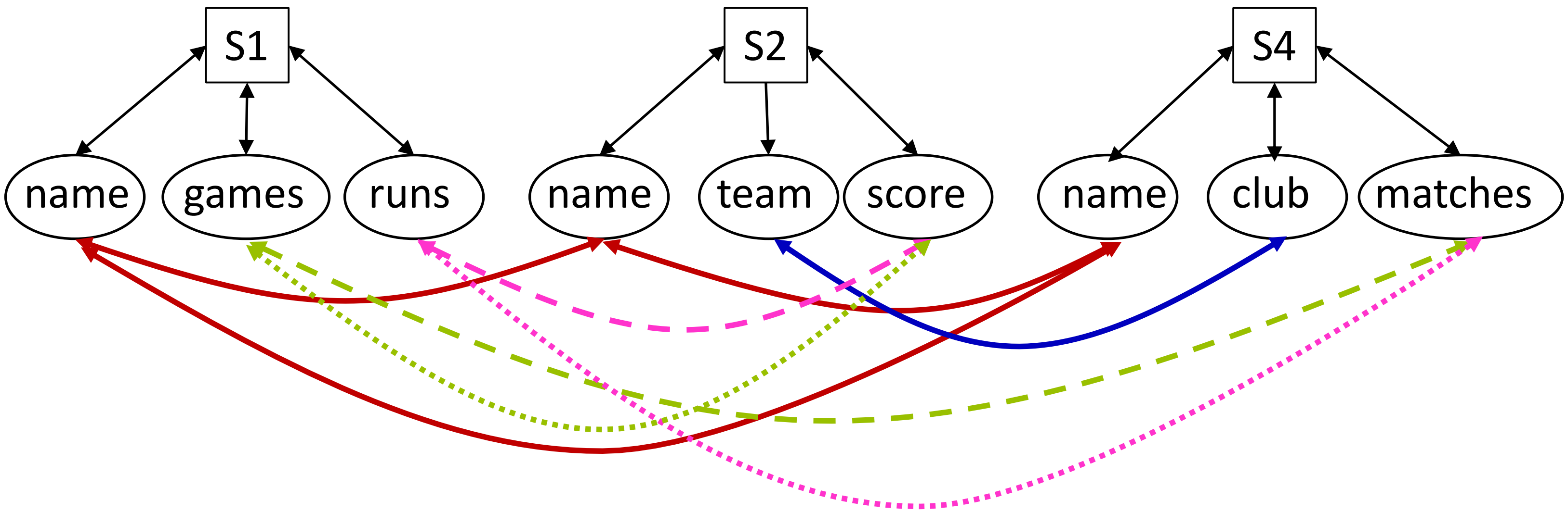
- Approach has worked for RDBMS, Web, Hadoop …

Probabilistic Mediated Schemas [DDH08]



- Mediated schemas: automatically created by inspecting sources
  - Clustering of source attributes
  - **Volume**, **variety** of sources → uncertainty in accuracy of clustering

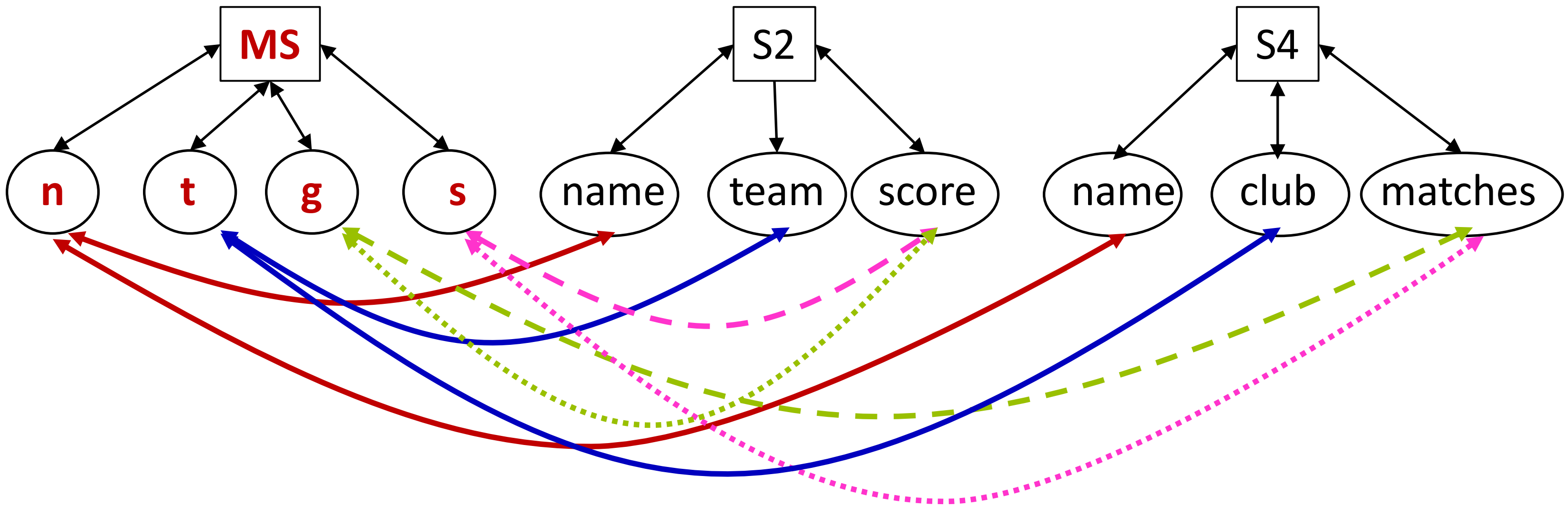
Probabilistic Mediated Schemas [DDH08]



- Example P-mediated schema
  - M1({S1.games, S4.matches}, {S1.runs, S2.score})
  - M2({S1.games, S2.score}, {S1.runs, S4.matches})
  - M = {(M1, 0.6), (M2, 0.2), (M3, 0.1), (M4, 0.1)}

Probabilistic Mappings [DHY07, DDH09]

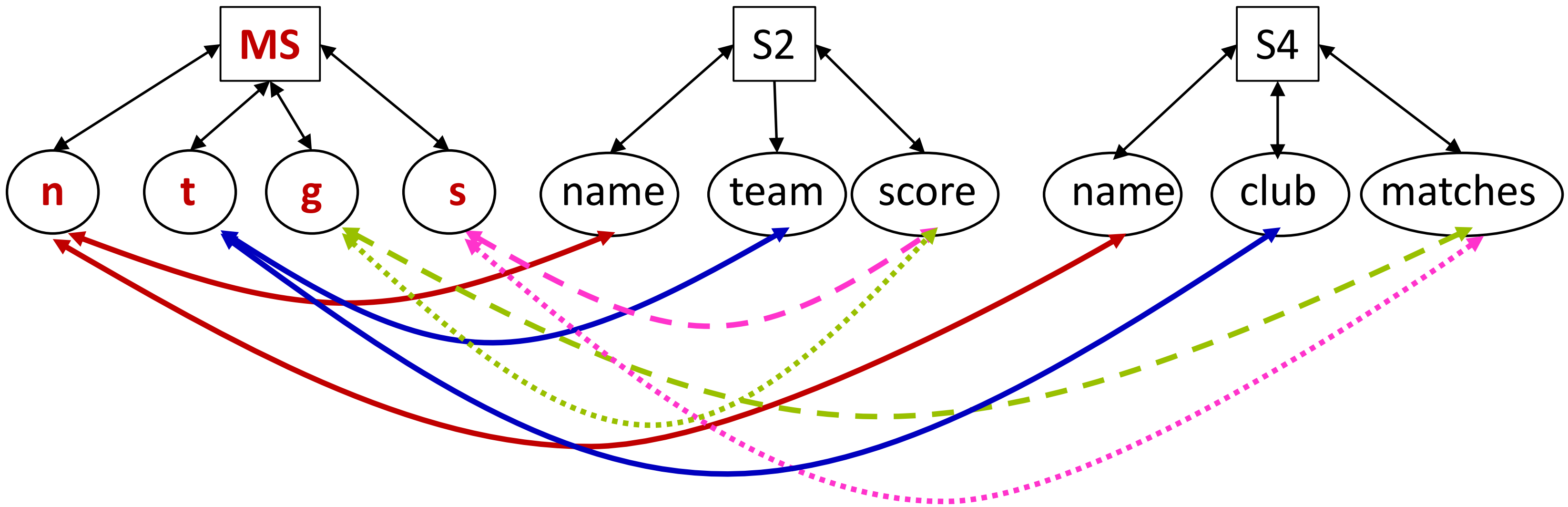
- Mapping between P-mediated and source schemas



- Example mappings
  - G1({**MS.t**, S2.team, S4.club}, {**MS.g**, S4.matches}, {**MS.s**, S2.score})
  - G2({**MS.t**, S2.team, S4.club}, {**MS.g**, S2.score}, {**MS.s**, S4.matches})
  - G = {(G1, 0.6), (G2, 0.2), (G3, 0.1), (G4, 0.1)}

Probabilistic Mappings [DHY07, DDH09]

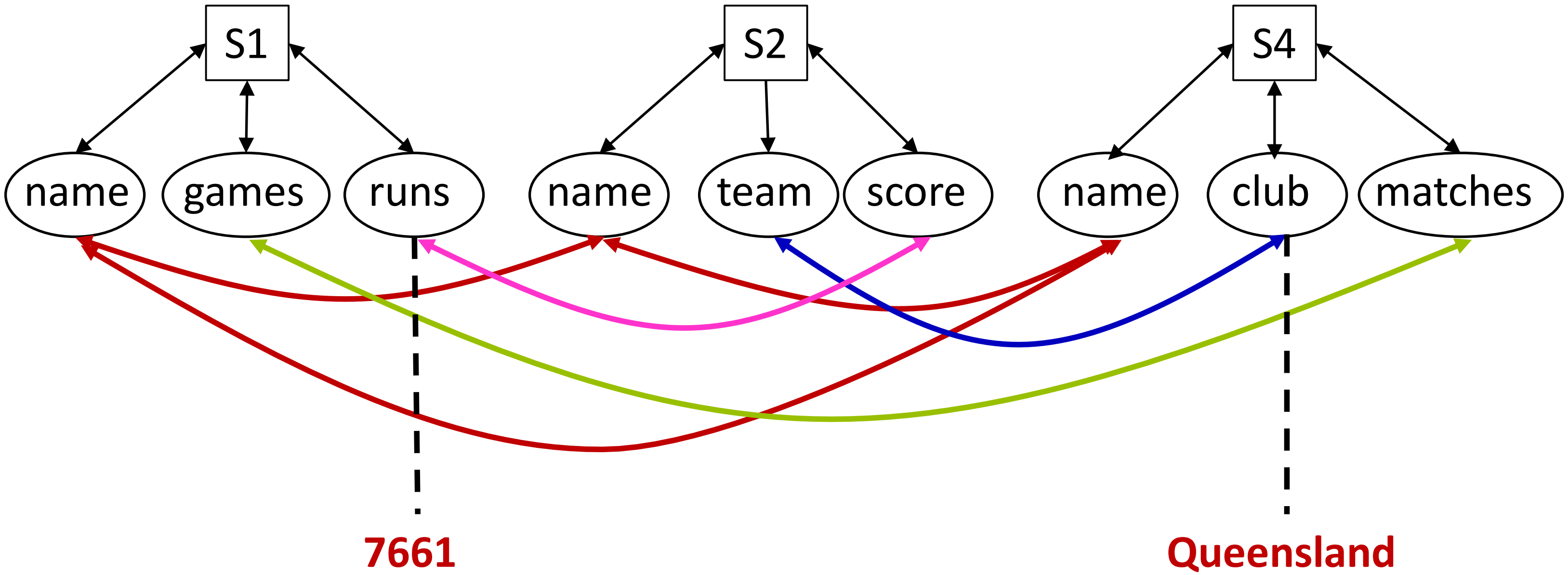
- Mapping between P-mediated and source schemas



- Answering queries on P-mediated schema based on P-mappings
  - By table semantics: one mapping is correct for all tuples
  - By tuple semantics: different mappings correct for different tuples

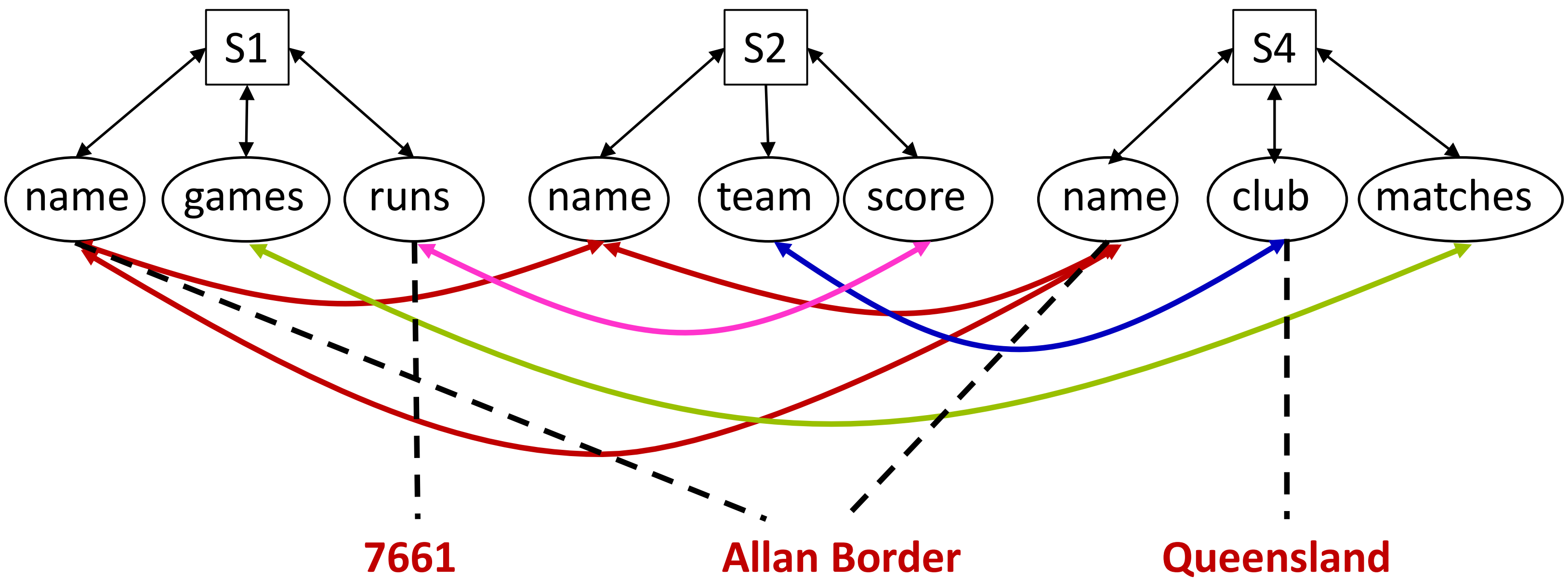
Keyword Search Based Integration [TJM+08]

- Key idea: information need driven integration
  - Search graph: source tables with weighted associations
  - Query keywords: matched to elements in different sources
  - Derive top-k SQL view, using Steiner tree on search graph



Keyword Search Based Integration [TJM+08]

- Key idea: information need driven integration
  - Search graph: source tables with weighted associations
  - Query keywords: matched to elements in different sources
  - Derive top-k SQL view, using Steiner tree on search graph

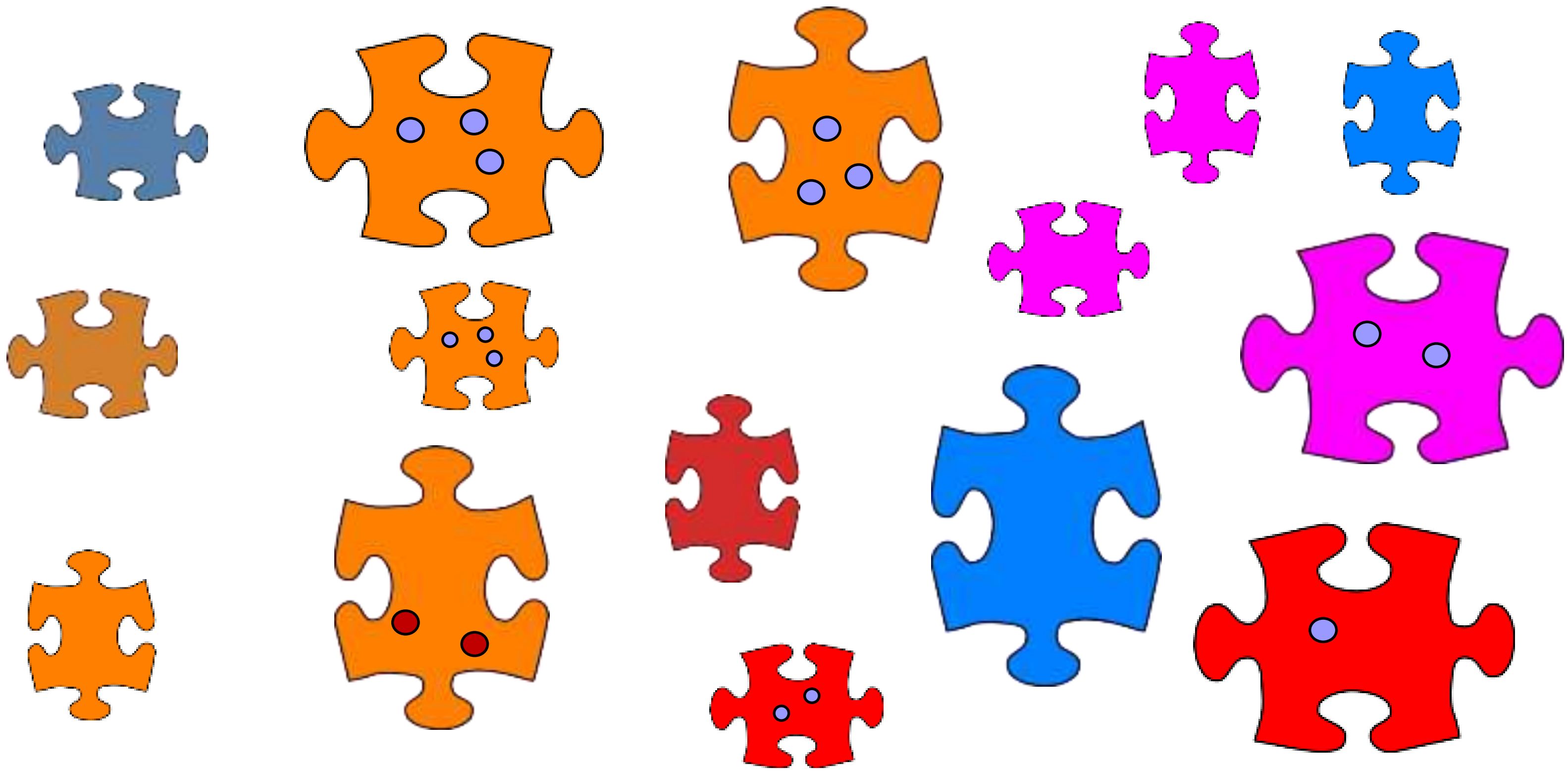


Outline

- Motivation
- Schema alignment
- Record linkage
  - Overview
  - Techniques for big data
- Data fusion

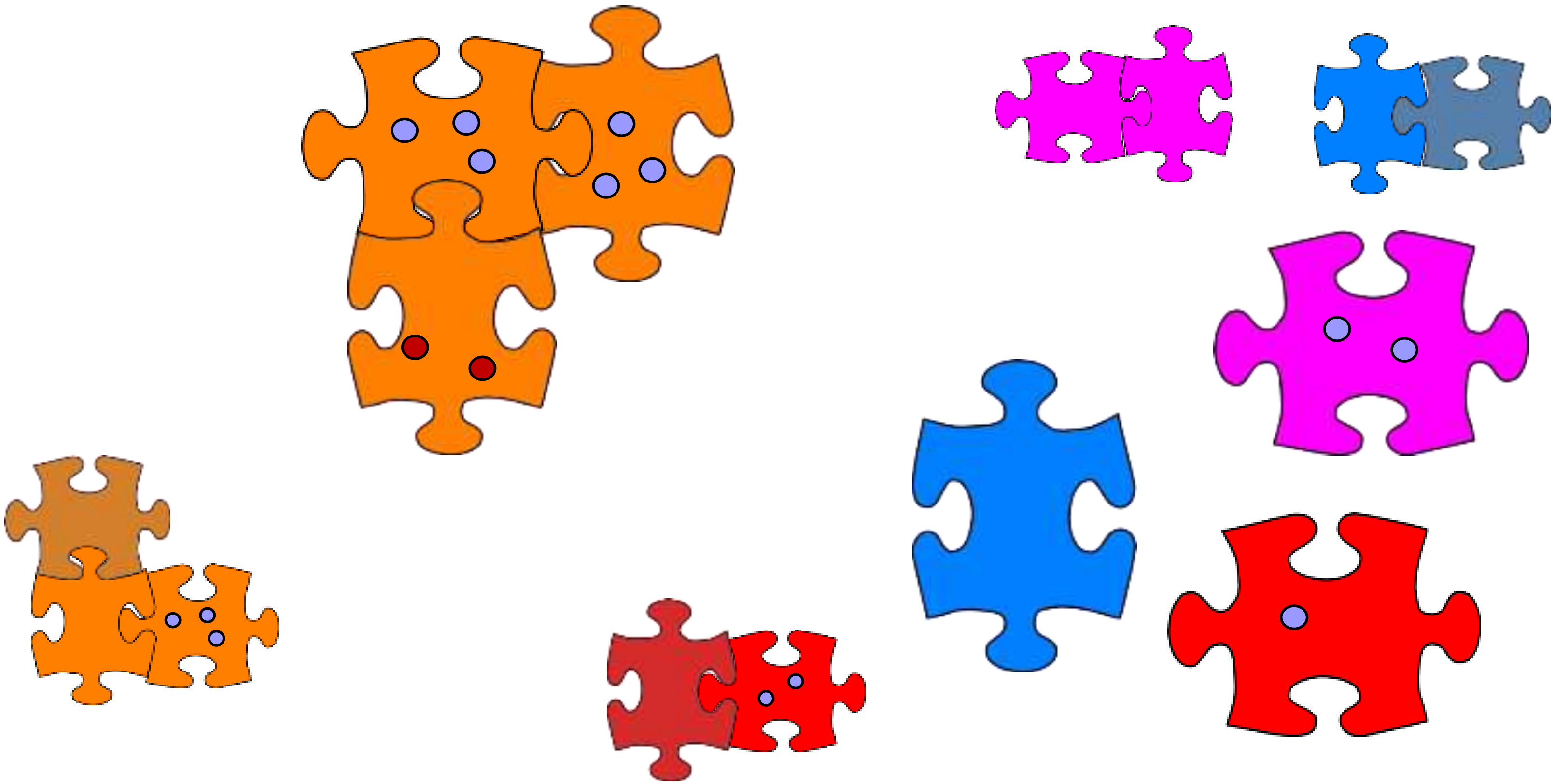
Record Linkage

- Matching based on **identifying** content: color, size



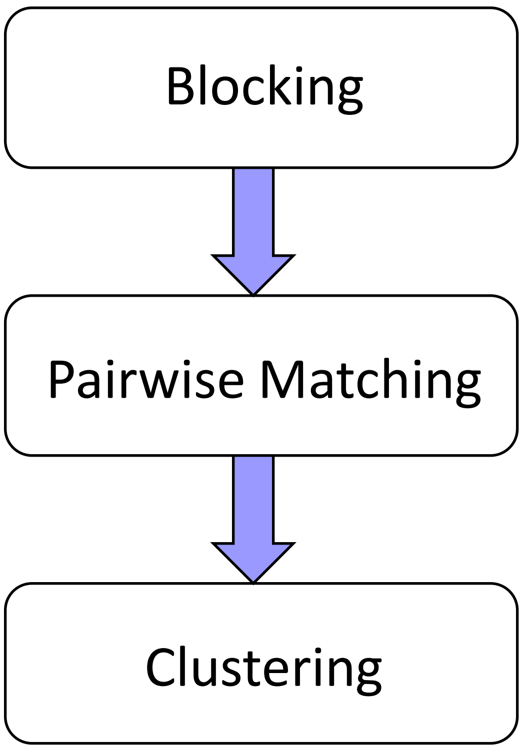
Record Linkage

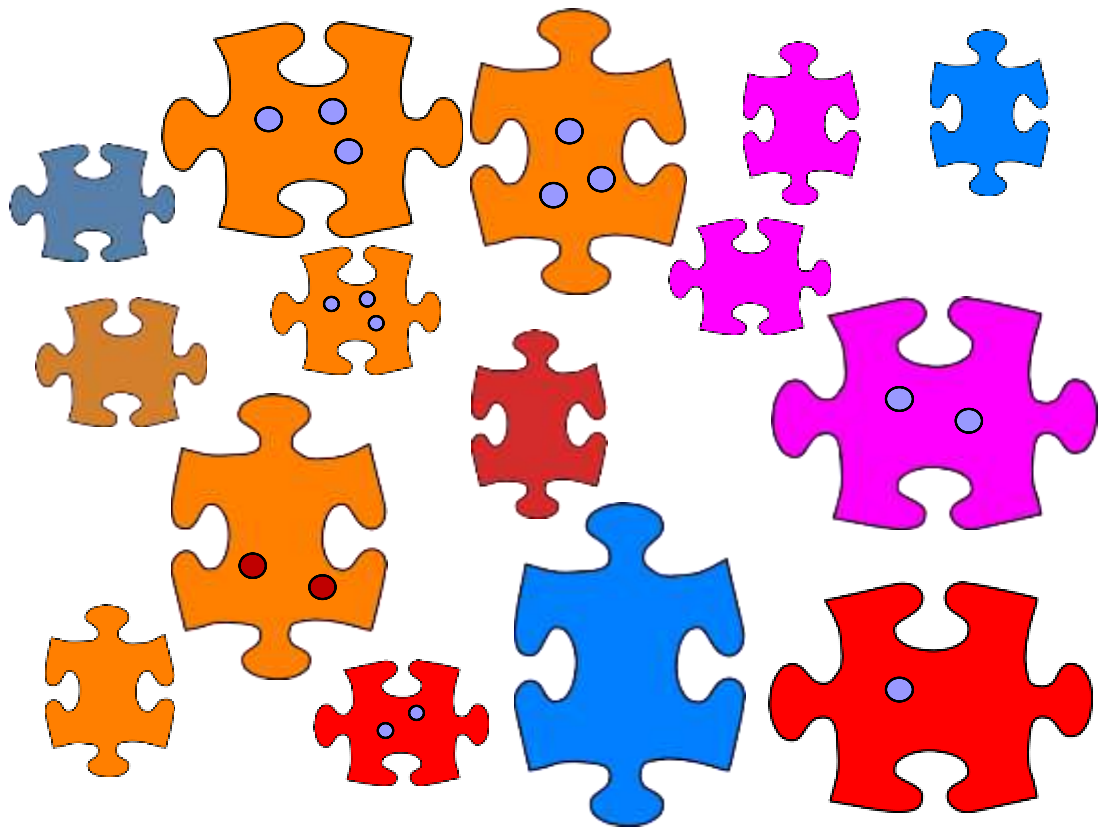
- Matching based on identifying content: color, size



Record Linkage: Three Steps [EIV07, GM12]

- Record linkage: blocking + pairwise matching + clustering
  - Scalability, similarity, semantics





Record Linkage: Three Steps

- Blocking: **efficiently** create **small** blocks of **similar** records
  - Ensures scalability

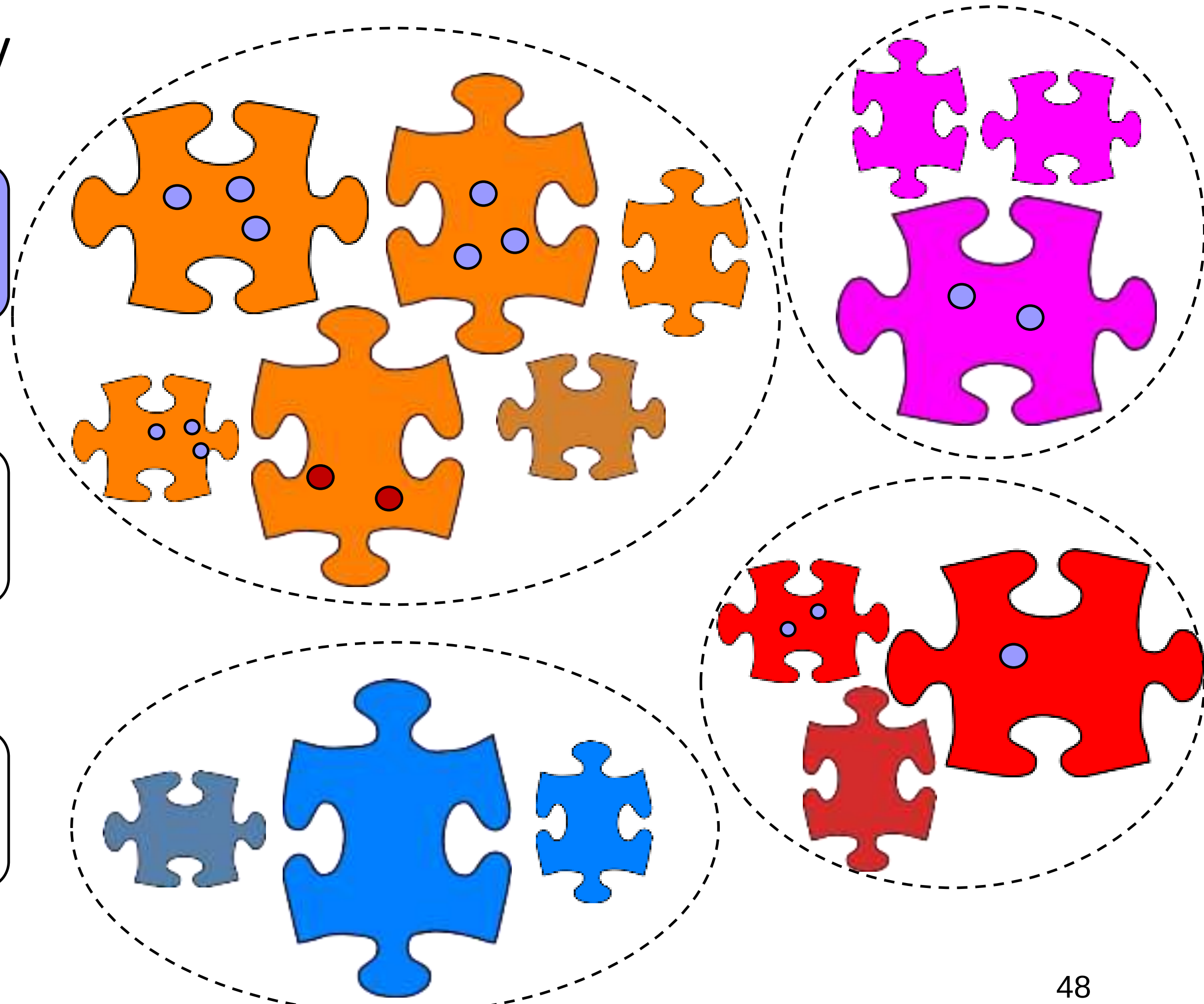
Blocking

↓

Pairwise Matching

↓

Clustering



Record Linkage: Three Steps

- Pairwise matching: compares all record pairs in a block
  - Computes similarity

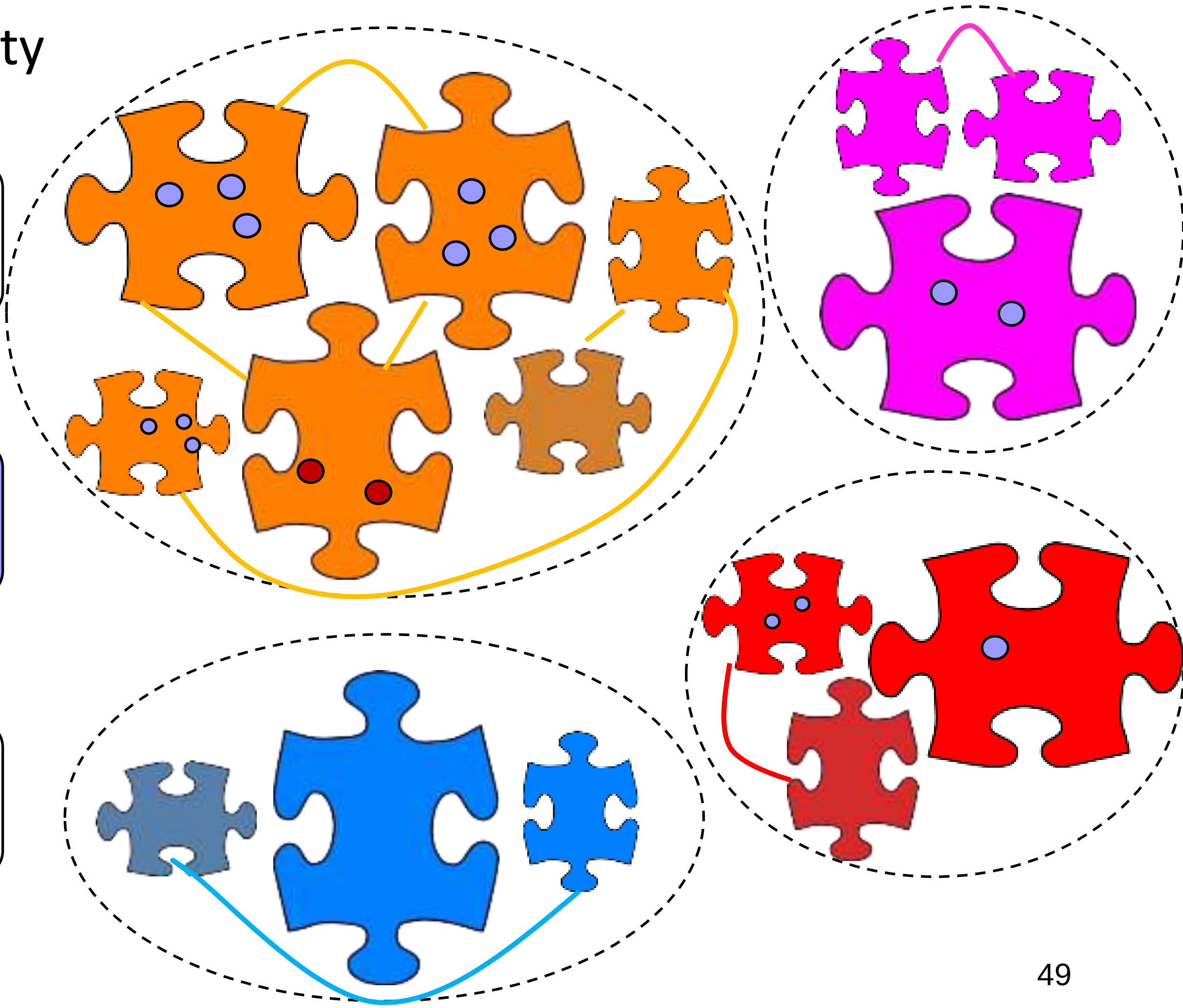
Blocking

↓

Pairwise Matching

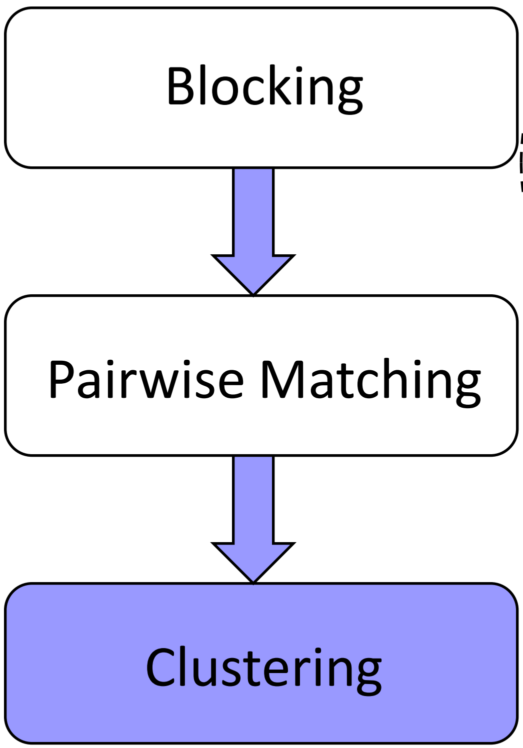
↓

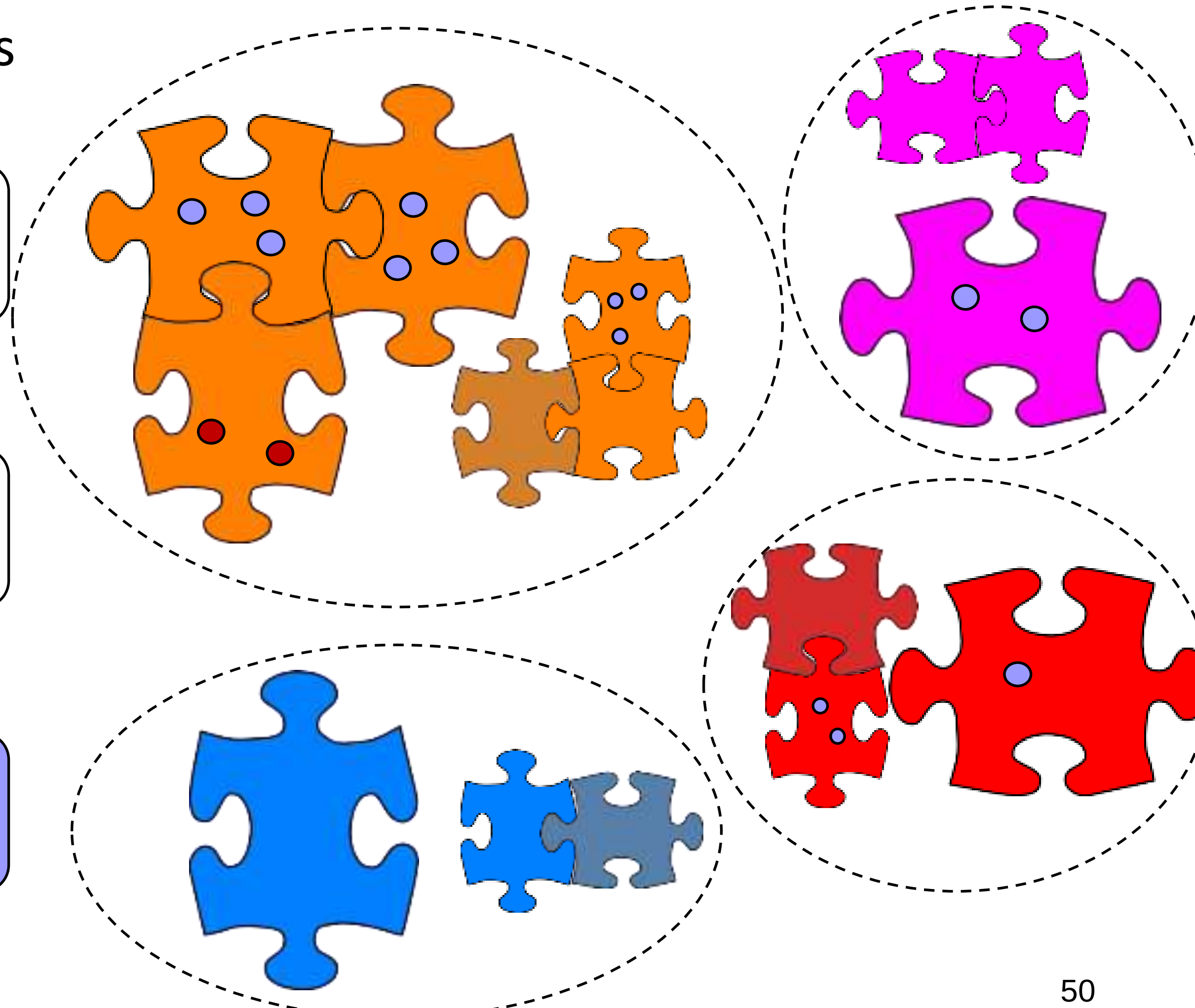
Clustering



Record Linkage: Three Steps

- Clustering: groups sets of records into entities
  - Ensures semantics





Outline

- Motivation
- Schema alignment
- Record linkage
  - Overview
  - Techniques for big data
- Data fusion

BDI: Record Linkage

- **Volume**: dealing with billions of records
  - Map-reduce based record linkage [VCL10, KTR12]
  - Adaptive record blocking [DNS+12, MKB12, VN12]
  - Blocking in heterogeneous data spaces [PIP+12]

- **Velocity**
  - Incremental record linkage [MSS10]

BDI: Record Linkage

- **Variety**
  - Matching structured and unstructured data [KGA+11, KTT+12]
- **Veracity**
  - Linking temporal records [LDM+11]

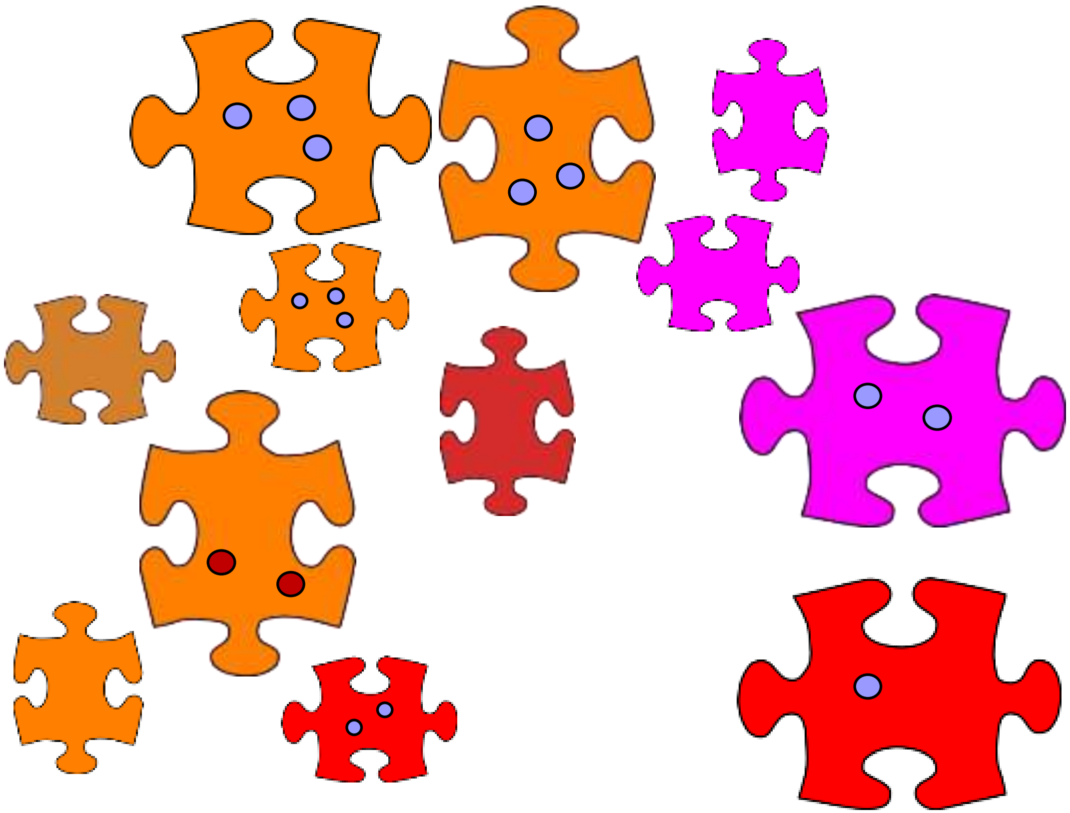
Record Linkage Using MapReduce [KTR12]

- Motivation: despite use of blocking, record linkage is expensive
  - Can record linkage be effectively parallelized?

- Basic: use MapReduce to execute blocking-based RL in parallel
  - **Map** tasks can read records, redistribute based on blocking key
  - All entities of the same block are assigned to same **Reduce** task
  - Different blocks matched in **parallel** by multiple Reduce tasks

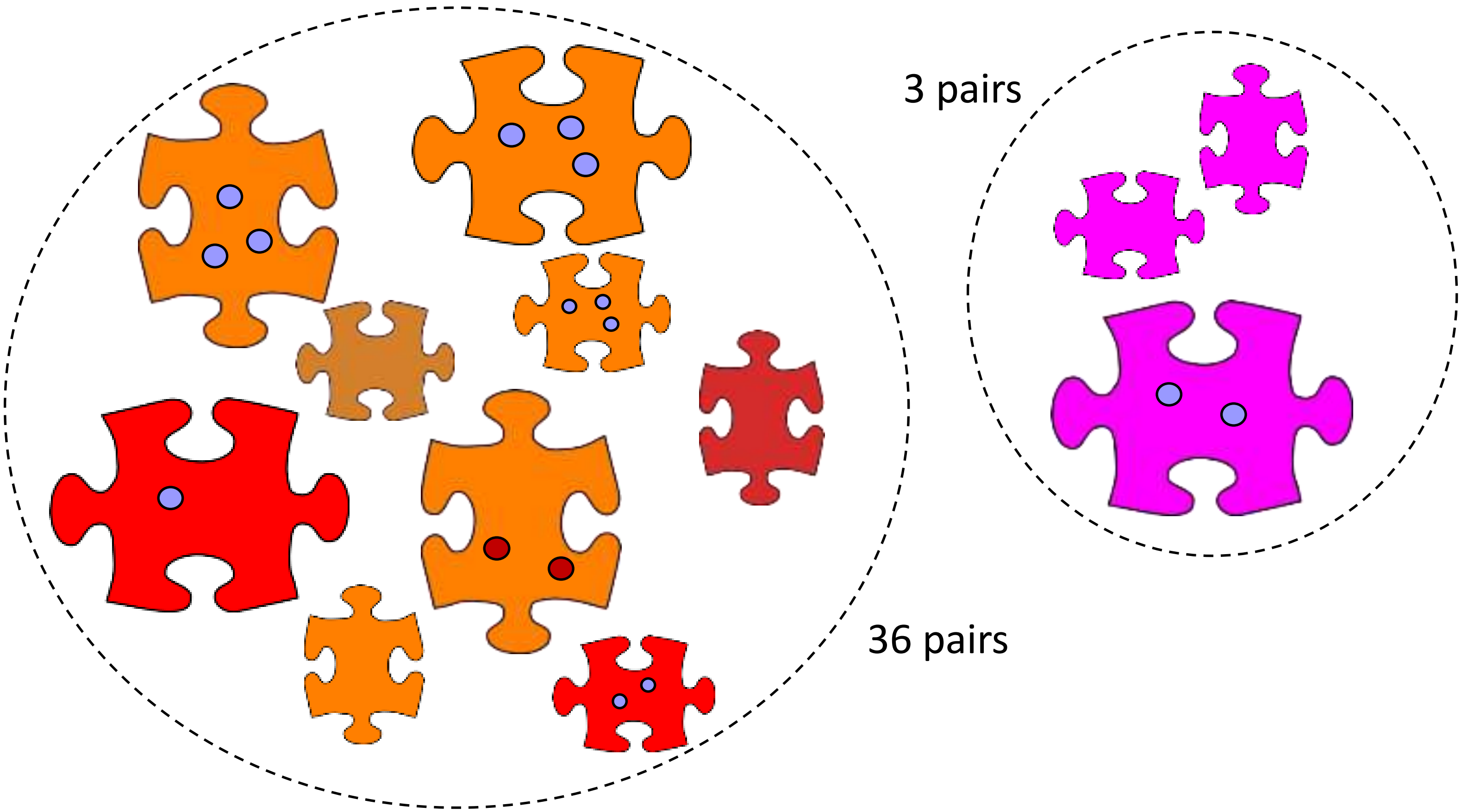
Record Linkage Using MapReduce

- Challenge: data skew → unbalanced workload



Record Linkage Using MapReduce

- Challenge: data skew → unbalanced workload
  - Speedup: 39/36 = 1.083



Load Balancing

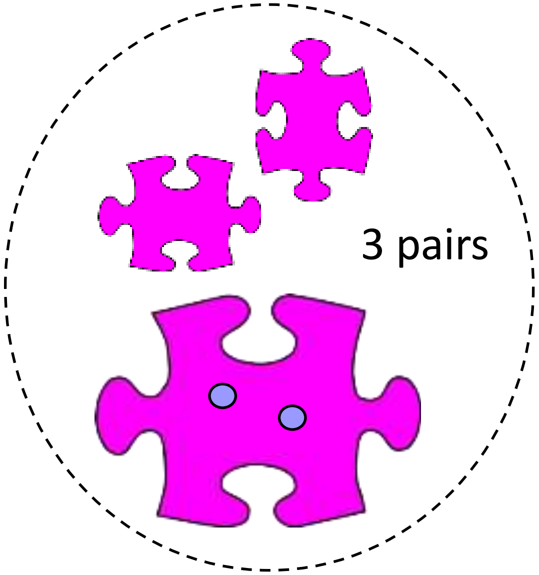
- Challenge: data skew → unbalanced workload
  - Difficult to tune blocking function to get balanced workload

- Key ideas for load balancing
  - **Preprocessing** MR job to determine blocking key distribution
  - Redistribution of **Match** tasks to **Reduce** tasks to balance workload

- Two load balancing strategies:
  - BlockSplit: split large blocks into sub-blocks
  - PairRange: global enumeration and redistribution of all pairs

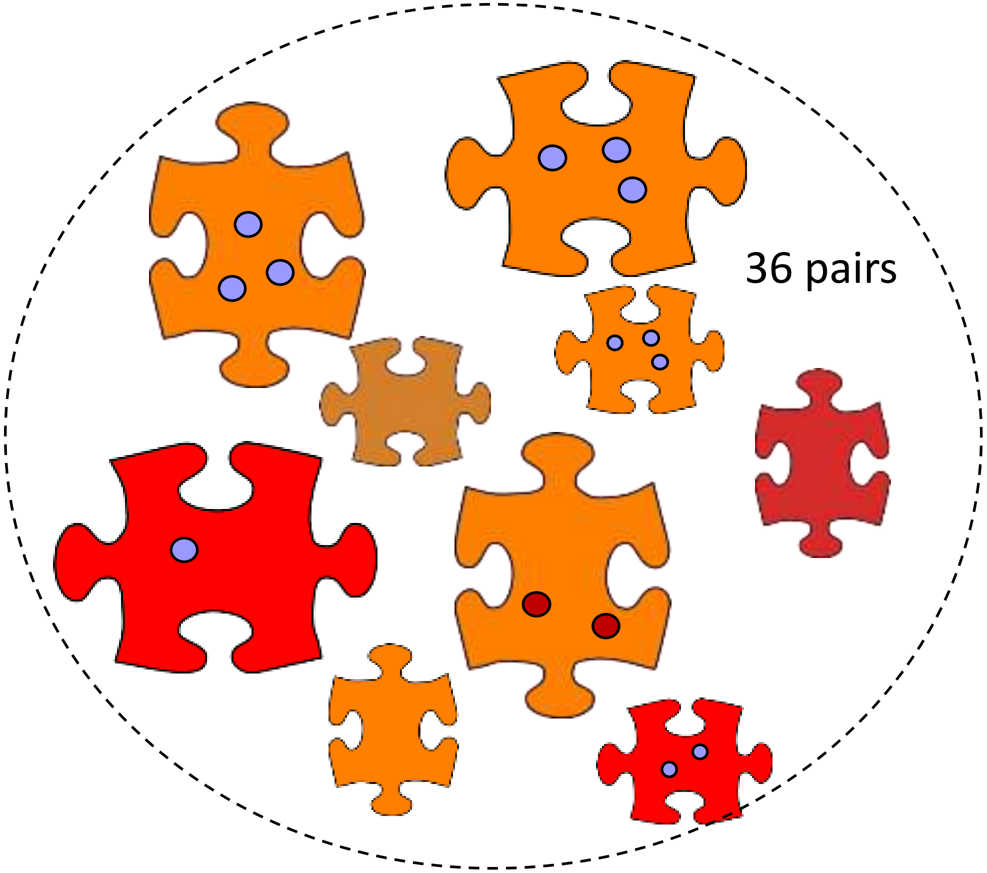
Load Balancing: BlockSplit

- Small blocks: processed by a single match task (as in Basic)



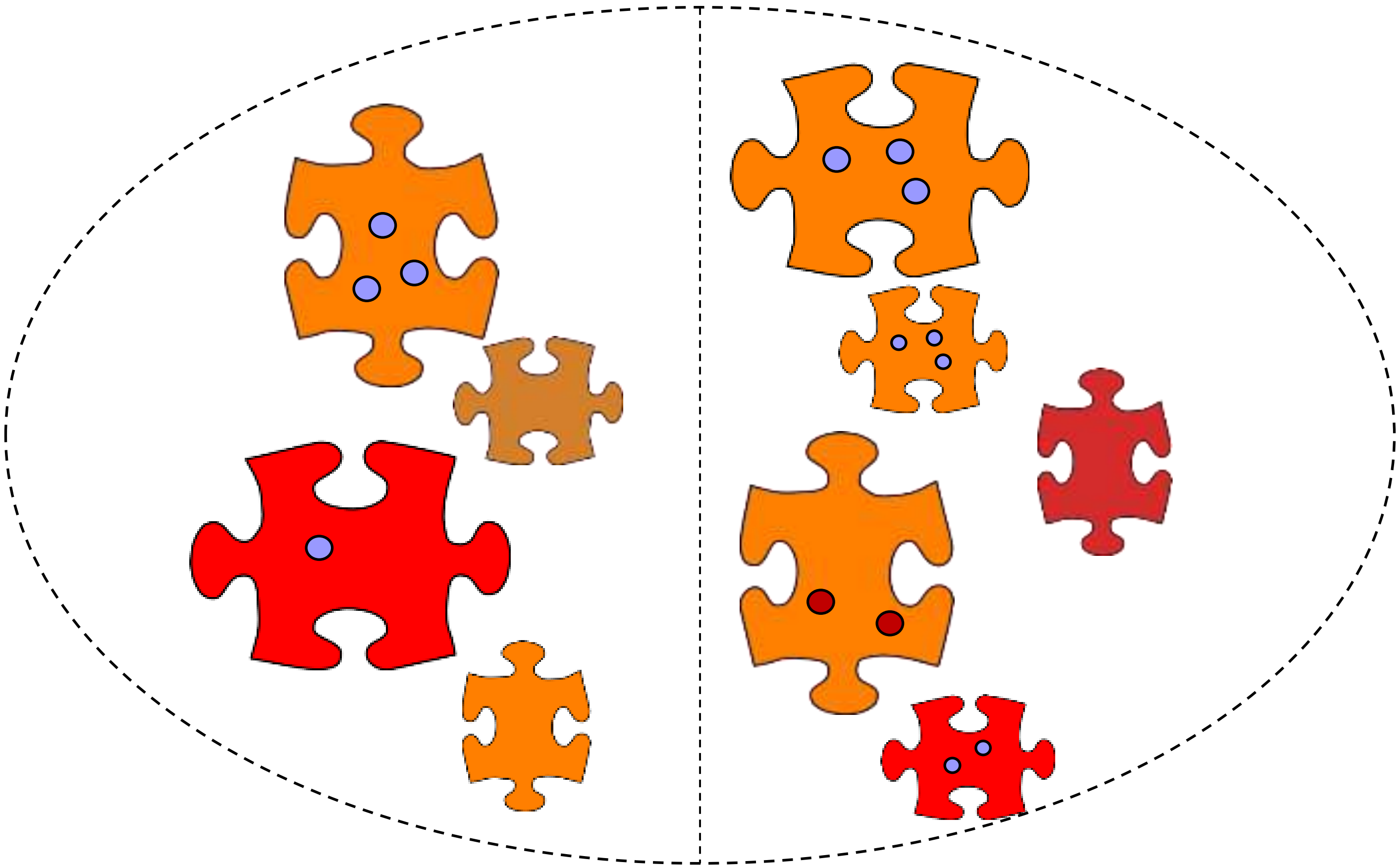
Load Balancing: BlockSplit

- Large blocks: split into multiple sub-blocks



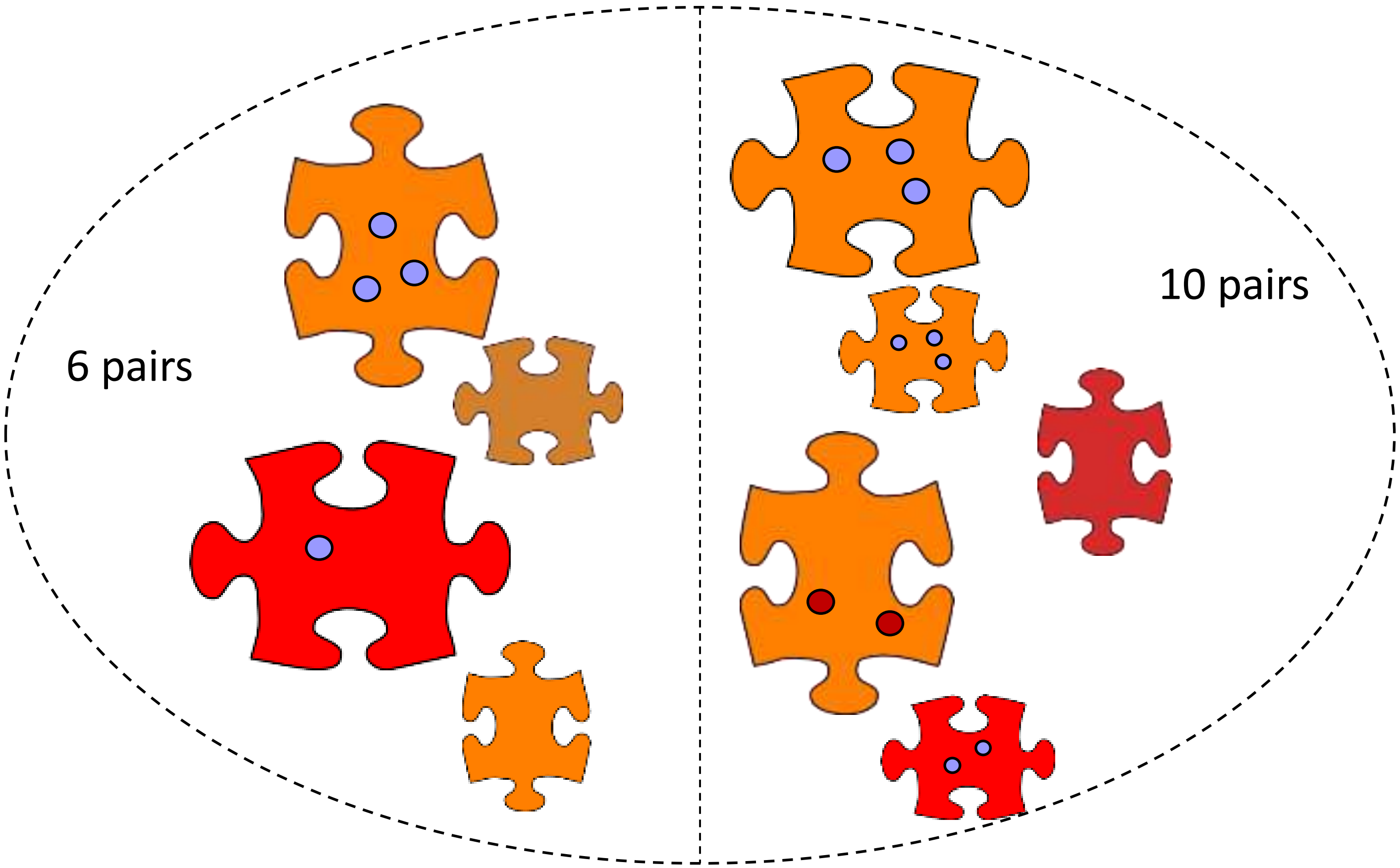
Load Balancing: BlockSplit

- Large blocks: split into multiple sub-blocks



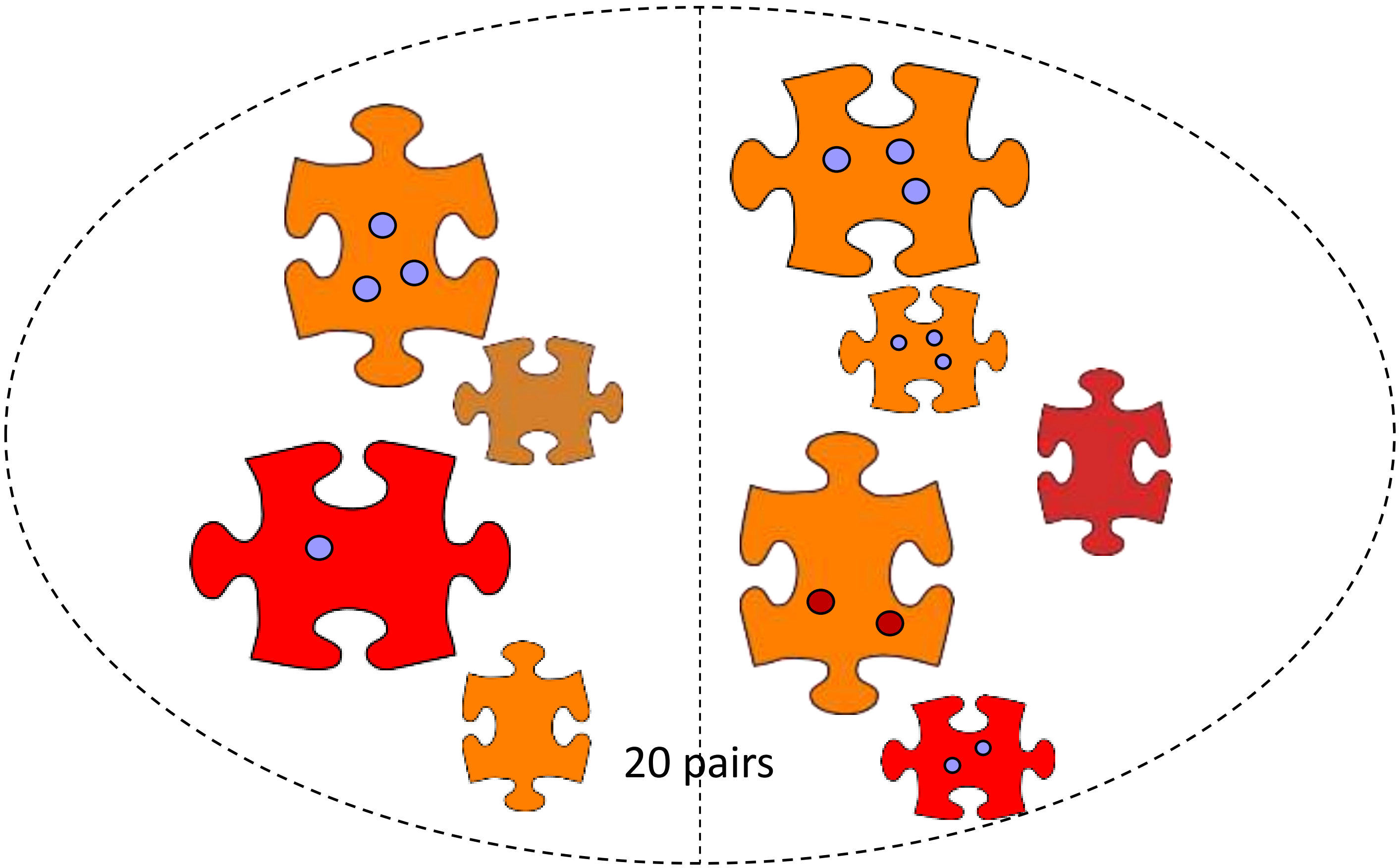
Load Balancing: BlockSplit

- Large blocks: split into multiple sub-blocks
  - Each sub-block processed (like unsplit block) by single match task



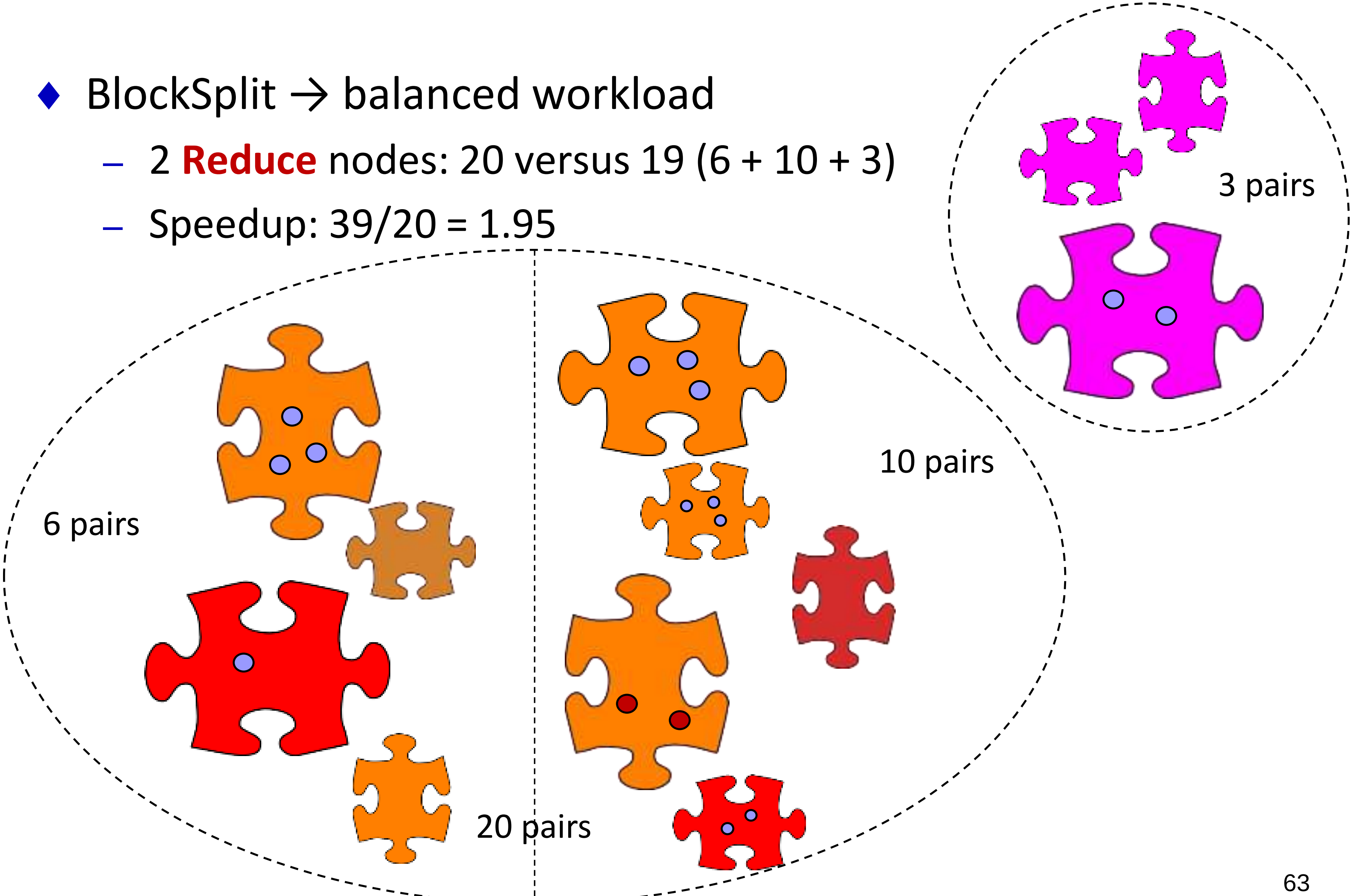
Load Balancing: BlockSplit

- Large blocks: split into multiple sub-blocks
  - Pair of sub-blocks is processed by “cartesian product” match task



Load Balancing: BlockSplit

- BlockSplit → balanced workload
  - 2 **Reduce** nodes: 20 versus 19 (6 + 10 + 3)
  - Speedup: 39/20 = 1.95



Structured + Unstructured Data [KGA+11]

- Motivation: matching offers to specifications with high precision
  - Product specifications are structured: set of (name, value) pairs
  - Product offers are terse, unstructured text
  - Many similar but different product offers, specifications

Attribute Name	Attribute Value
category	digital camera
brand	Panasonic
product line	Panasonic Lumix
model	DMC-FX07
resolution	7 megapixel
color	silver

Panasonic Lumix DMC-FX07 digital camera [7.2 megapixel, 2.5”, 3.6x , LCD monitor]

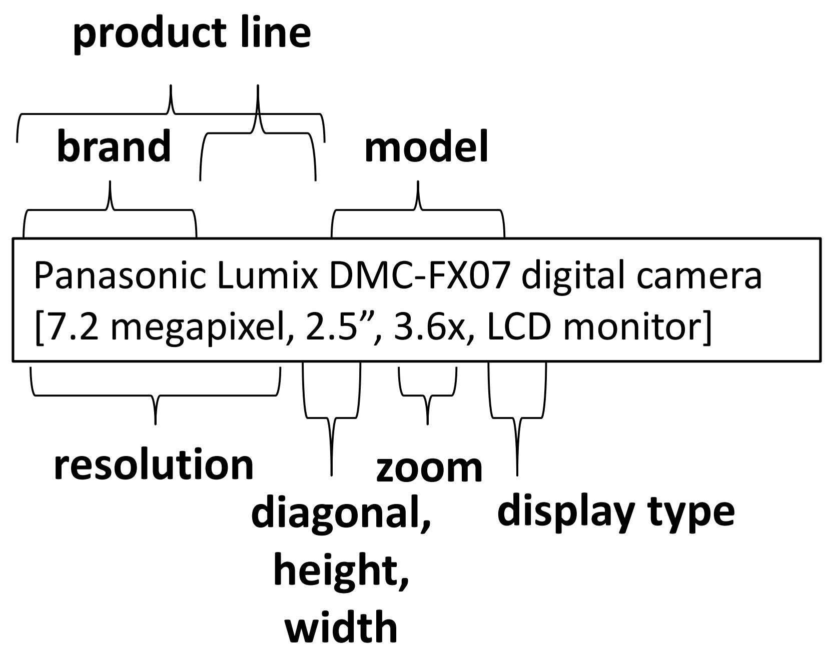
Panasonic DMC-FX07EB digital camera silver

Lumix FX07EB-S, 7.2MP

Structured + Unstructured Data

- Key idea: optimal parse of (unstructured) offer wrt specification

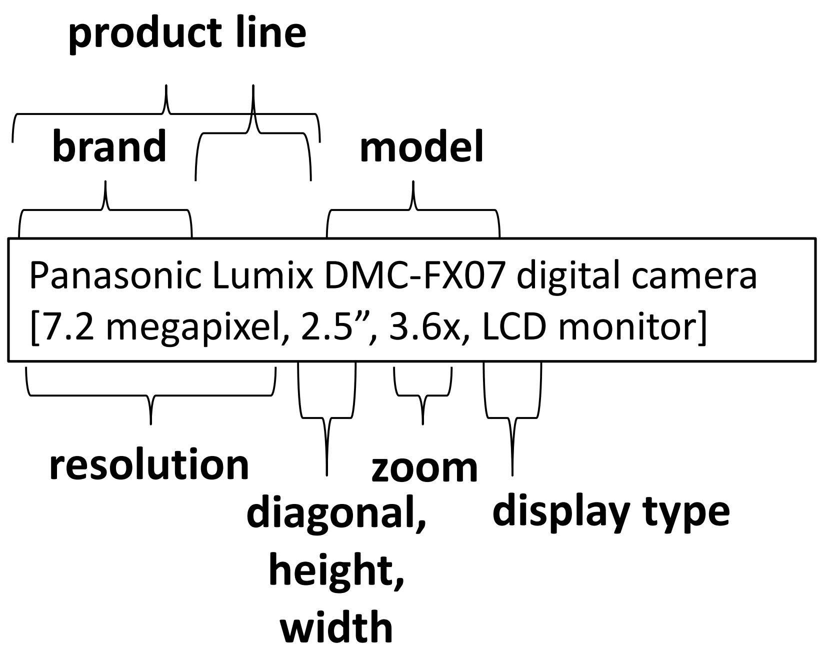
- Semantic parse of offers: tagging
  - Use inverted index built on specification values
  - Tag all n-grams



Structured + Unstructured Data

- Key idea: optimal parse of (unstructured) offer wrt specification

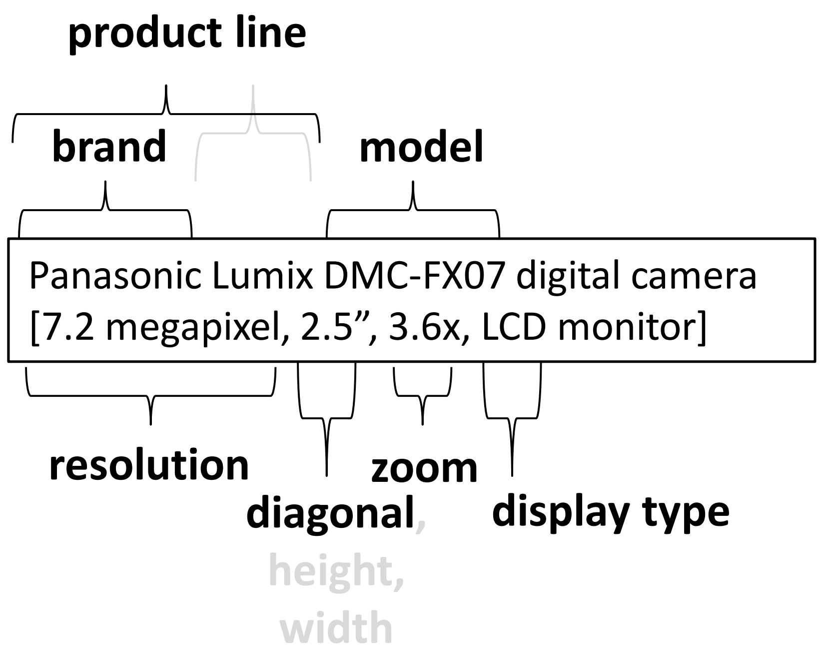
- Semantic parse of offers: tagging, plausible parse
  - Combination of tags such that each attribute has distinct value



Structured + Unstructured Data

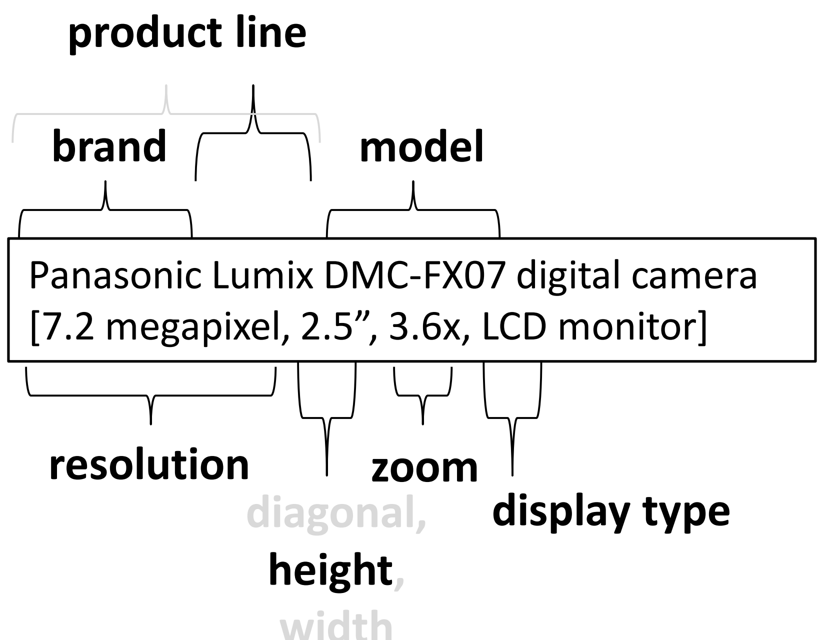
- Key idea: optimal parse of (unstructured) offer wrt specification

- Semantic parse of offers: tagging, plausible parse
  - Combination of tags such that each attribute has distinct value



Structured + Unstructured Data

- Key idea: optimal parse of (unstructured) offer wrt specification
- Semantic parse of offers: tagging, plausible parse
  - Combination of tags such that each attribute has distinct value
  - # depends on ambiguities



Structured + Unstructured Data

- Key idea: optimal parse of (unstructured) offer wrt specification

- Semantic parse of offers: tagging, plausible parse, optimal parse
  - Optimal parse depends on the product specification

Product specification		Optimal Parse
brand product line model diagonal	Panasonic Lumix DMC-FX05 2.5 in	Panasonic Lumix DMC-FX07 digital camera [7.2 megapixel, 2.5", 3.6x, LCD monitor]
brand model resolution zoom	Panasonic DMC-FX07 7.2 megapixel 3.6x	Panasonic Lumix DMC-FX07 digital camera [7.2 megapixel, 2.5", 3.6x, LCD monitor]

Structured + Unstructured Data

- Key idea: optimal parse of (unstructured) offer wrt specification

- Semantic parse of offers: tagging, plausible parse, optimal parse

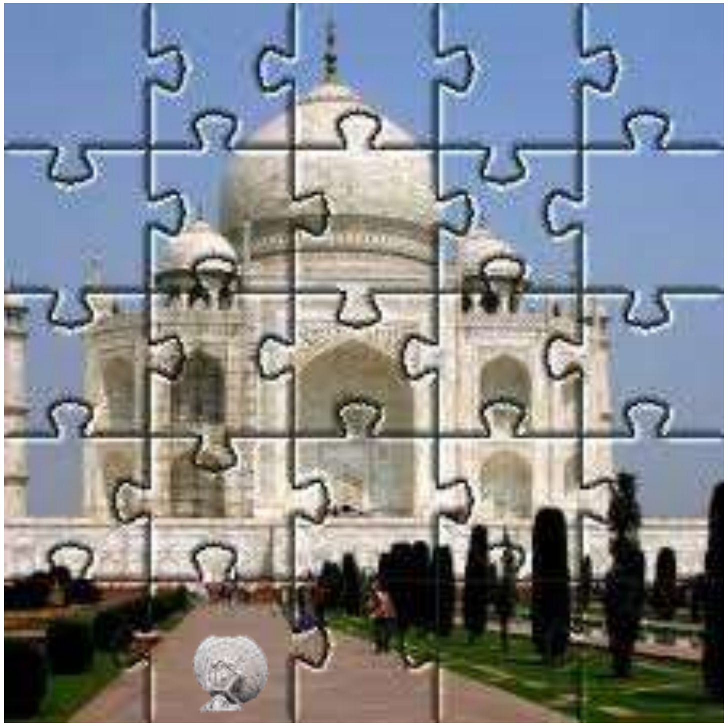
- Finding specification with largest match probability is now easy
  - Similarity feature vector between offer and specification: {-1, 0, 1}*
  - Use binary logistic regression to learn weights of each feature
  - Blocking 1: use classifier to categorize offer into product category
  - Blocking 2: identify candidates with ≥ 1 high weighted feature

Outline

- Motivation
- Schema alignment
- Record linkage
- Data fusion
  - Overview
  - Techniques for big data

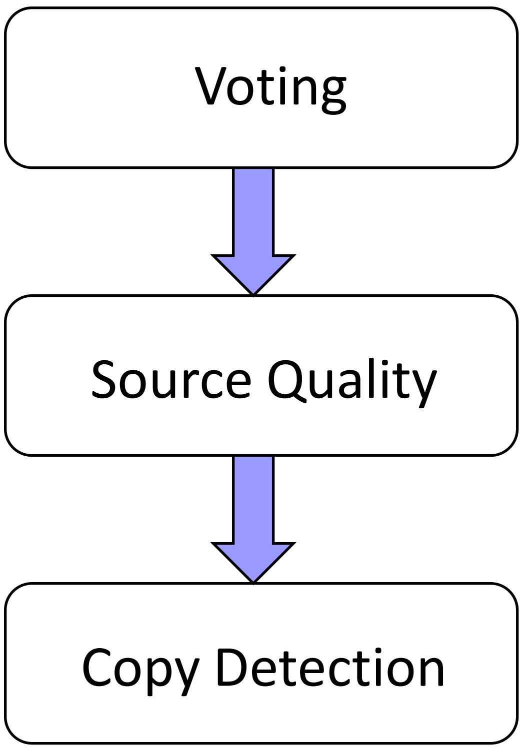
Data Fusion

- Reconciliation of conflicting non-identifying content



Data Fusion: Three Components [DBS09a]

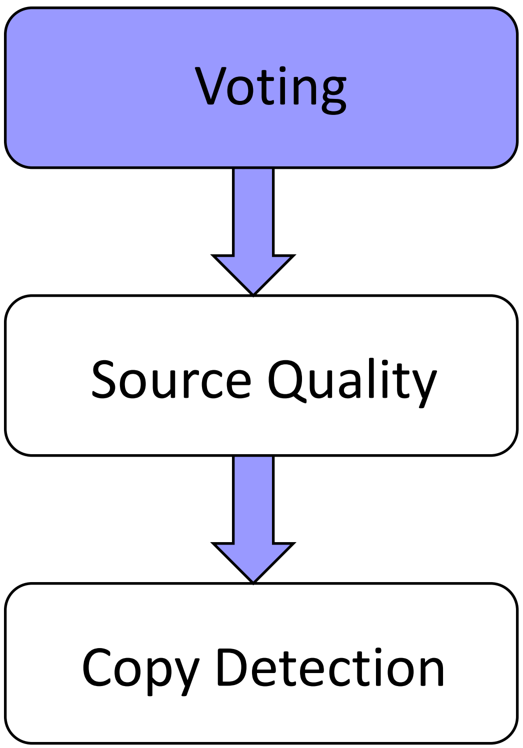
- Data fusion: voting + source quality + copy detection
  - Resolves inconsistency across diversity of sources



	S1	S2	S3	S4	S5
Jagadish	UM	ATT	UM	UM	UI
Dewitt	MSR	MSR	UW	UW	UW
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	ATT	BEA	BEA	BEA
Franklin	UCB	UCB	UMD	UMD	UMD

Data Fusion: Three Components [DBS09a]

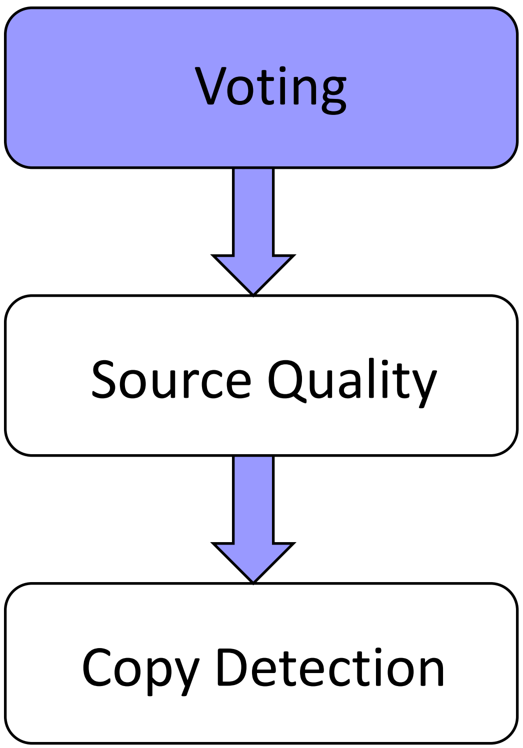
- Data fusion: voting + source quality + copy detection



	S1	S2	S3
Jagadish	UM	ATT	UM
Dewitt	MSR	MSR	UW
Bernstein	MSR	MSR	MSR
Carey	UCI	ATT	BEA
Franklin	UCB	UCB	UMD

Data Fusion: Three Components [DBS09a]

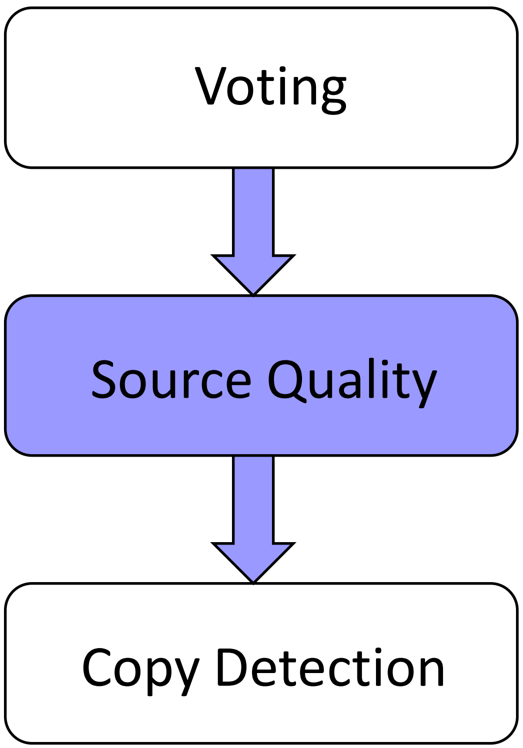
- Data fusion: voting + source quality + copy detection
  - Supports difference of opinion



	S1	S2	S3
Jagadish	**UM**	ATT	**UM**
Dewitt	**MSR**	**MSR**	UW
Bernstein	**MSR**	**MSR**	**MSR**
Carey	UCI	ATT	BEA
Franklin	**UCB**	**UCB**	UMD

Data Fusion: Three Components [DBS09a]

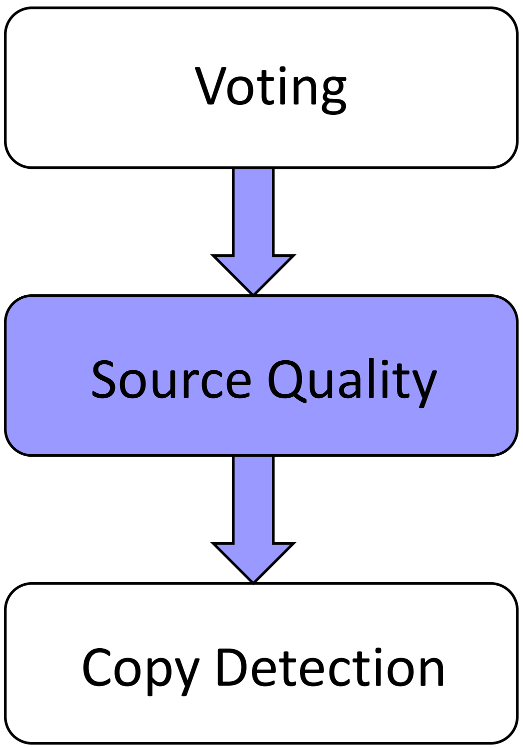
- Data fusion: voting + source quality + copy detection



	S1	S2	S3
Jagadish	**UM**	ATT	**UM**
Dewitt	**MSR**	**MSR**	UW
Bernstein	**MSR**	**MSR**	**MSR**
Carey	UCI	ATT	BEA
Franklin	**UCB**	**UCB**	UMD

Data Fusion: Three Components [DBS09a]

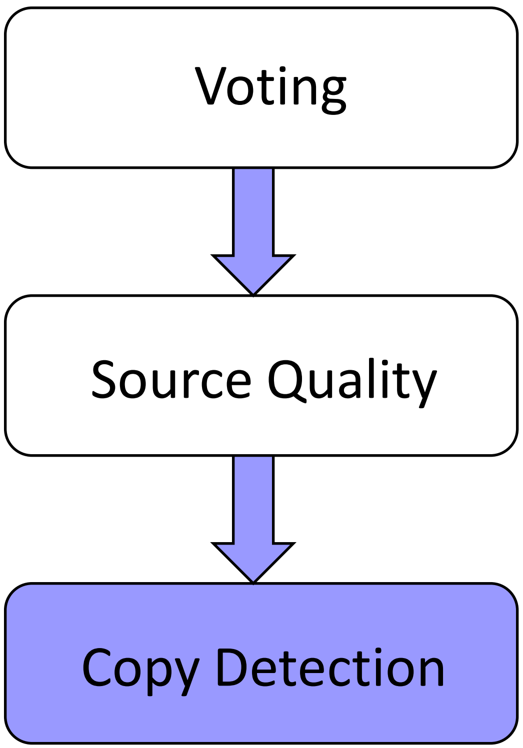
- Data fusion: voting + source quality + copy detection
  - Gives more weight to knowledgeable sources



	S1	S2	S3
Jagadish	**UM**	ATT	**UM**
Dewitt	**MSR**	**MSR**	UW
Bernstein	**MSR**	**MSR**	**MSR**
Carey	**UCI**	ATT	BEA
Franklin	**UCB**	**UCB**	UMD

Data Fusion: Three Components [DBS09a]

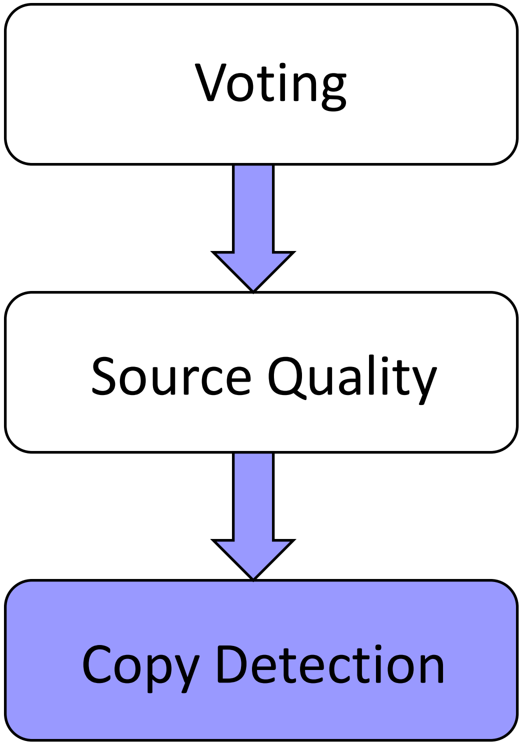
- Data fusion: voting + source quality + copy detection



	S1	S2	S3	S4	S5
Jagadish	UM	ATT	UM	UM	UI
Dewitt	MSR	MSR	UW	UW	UW
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	ATT	BEA	BEA	BEA
Franklin	UCB	UCB	UMD	UMD	UMD

Data Fusion: Three Components [DBS09a]

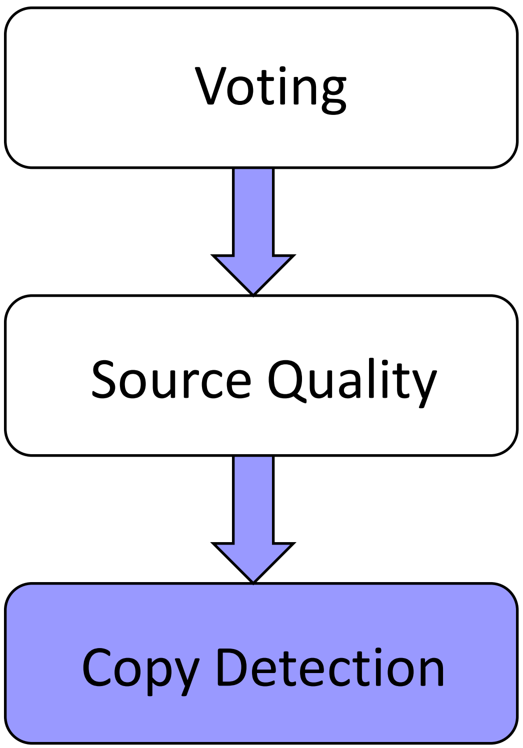
- Data fusion: voting + source quality + copy detection



	S1	S2	S3	S4	S5
Jagadish	UM	ATT	UM	UM	UI
Dewitt	MSR	MSR	UW	UW	UW
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	ATT	BEA	BEA	BEA
Franklin	UCB	UCB	UMD	UMD	UMD

Data Fusion: Three Components [DBS09a]

- Data fusion: voting + source quality + copy detection
  - Reduces weight of copier sources



	S1	S2	S3	S4	S5
Jagadish	UM	ATT	UM	UM	UI
Dewitt	MSR	MSR	UW	UW	UW
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	ATT	BEA	BEA	BEA
Franklin	UCB	UCB	UMD	UMD	UMD

Outline

- Motivation
- Schema alignment
- Record linkage
- Data fusion
  - Overview
  - Techniques for big data

BDI: Data Fusion

- **Veracity**
  - Using source trustworthiness [YJY08, GAM+10, PR11]
  - Combining source accuracy and copy detection [DBS09a]
  - Multiple truth values [ZRG+12]
  - Erroneous numeric data [ZH12]
  - Experimental comparison on deep web data [LDL+13]

BDI: Data Fusion

- **Volume**:
  - Online data fusion [LDO+11]
- **Velocity**
  - Truth discovery for dynamic data [DBS09b, PRM+12]
- **Variety**
  - Combining record linkage with data fusion [GDS+10]

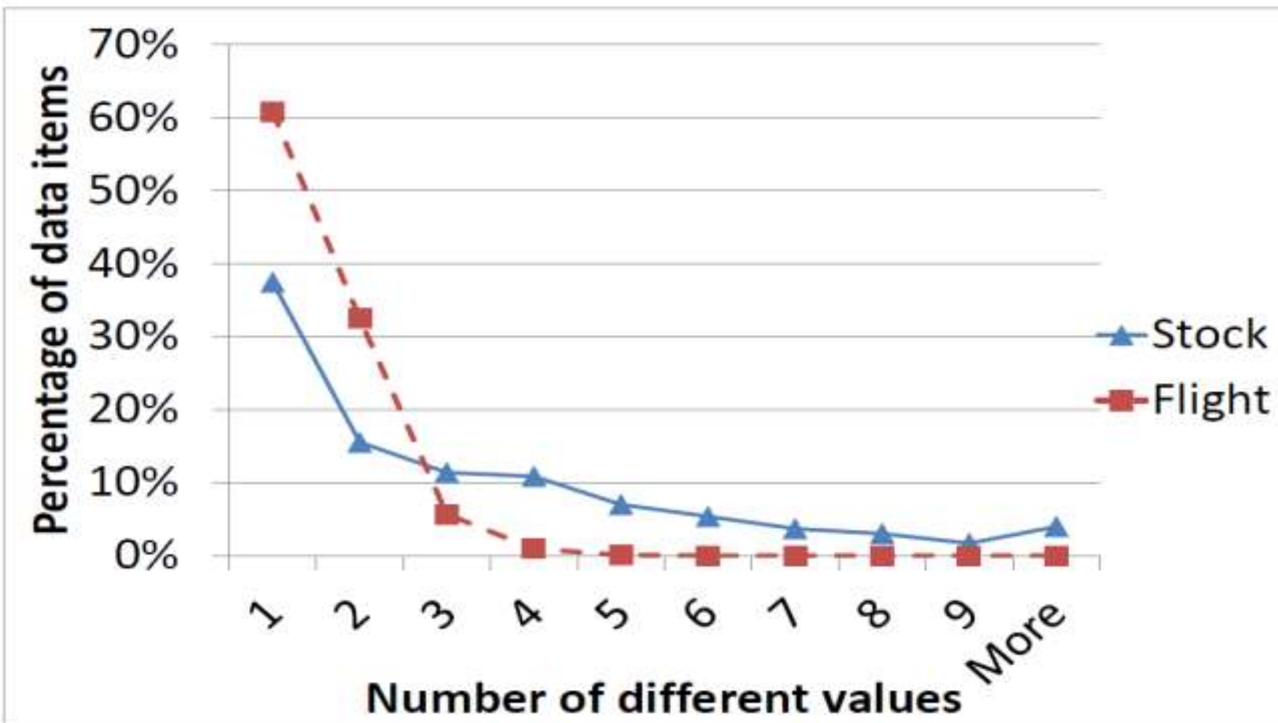
Experimental Study on Deep Web [LDL+13]

- Study on two domains
  - Belief of clean data
  - Poor quality data can have big impact

	#Sources	Period	#Objects	#Local-attrs	#Global-attrs	Considered items
Stock	55	7/2011	1000*20	333	153	16000*20
Flight	38	12/2011	1200*31	43	15	7200*31

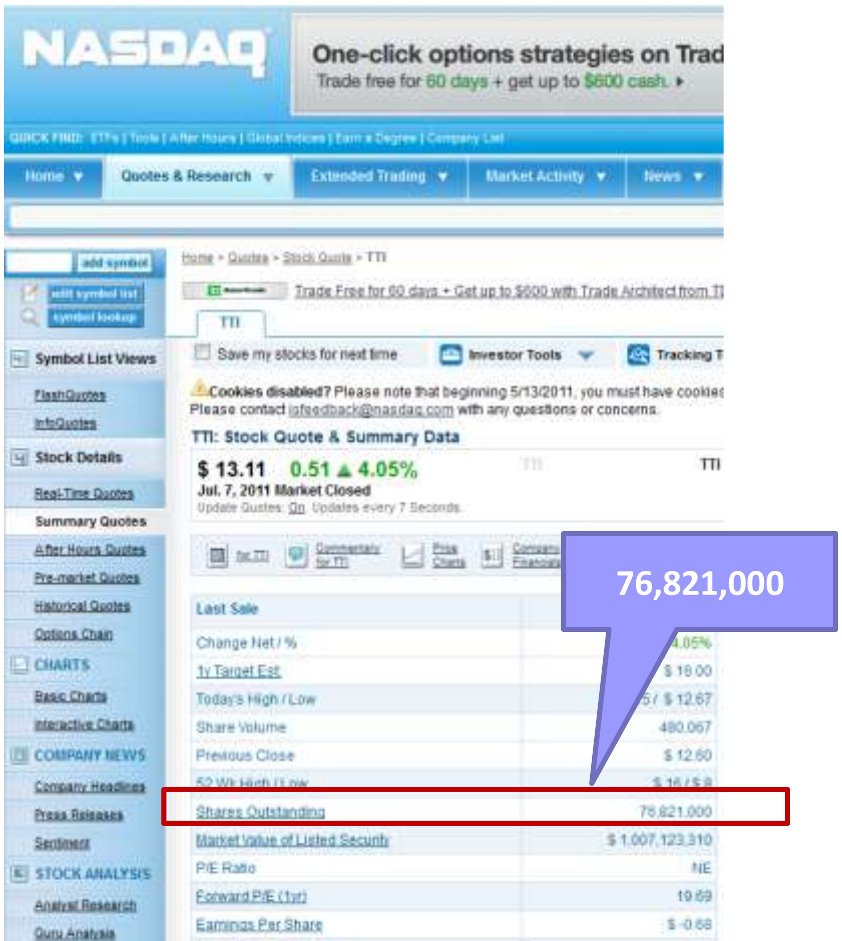
Experimental Study on Deep Web

- Is the data consistent?
  - Tolerance to 1% value difference



Experimental Study on Deep Web

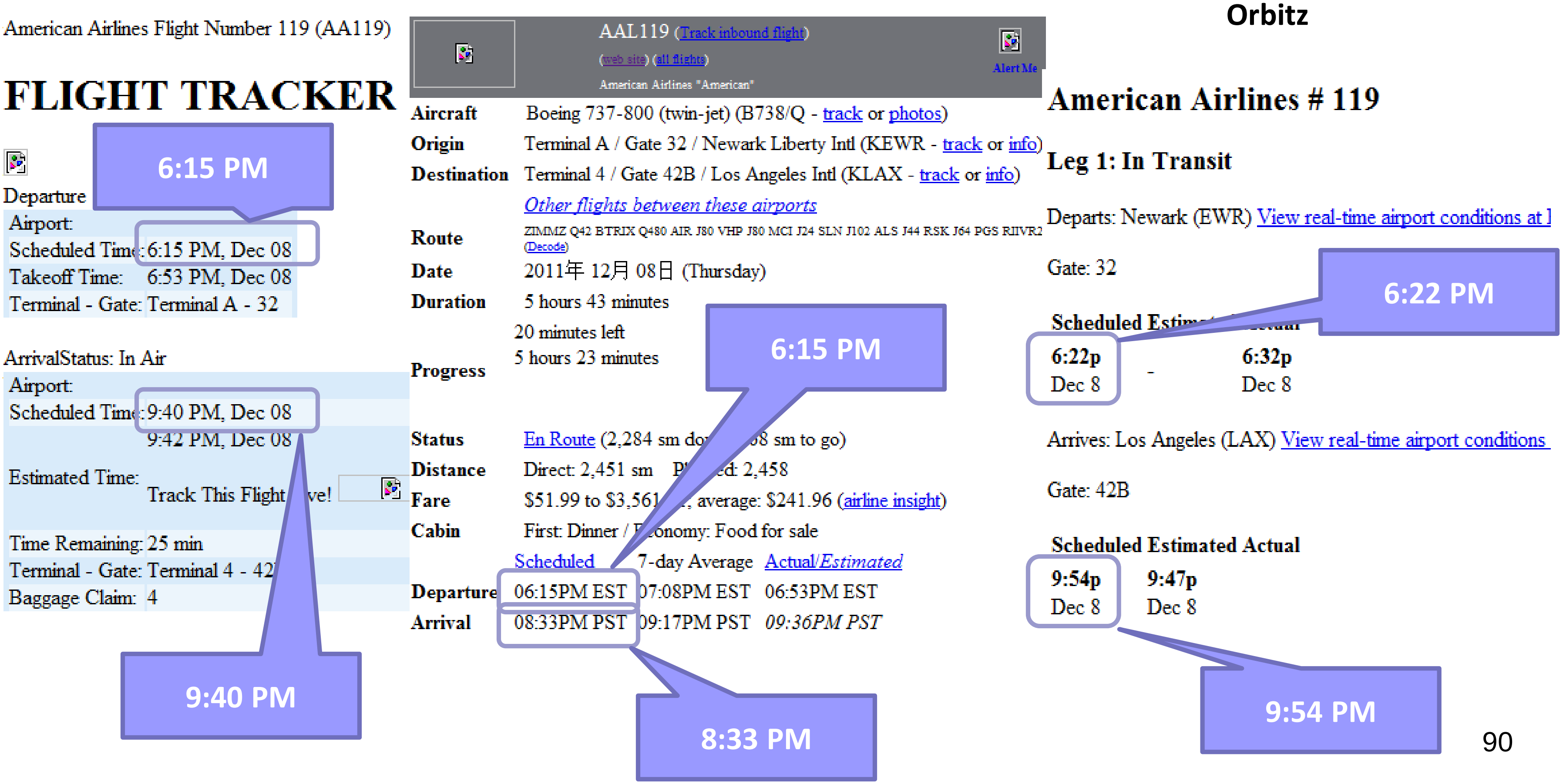
- Why such inconsistency?
  - Unit errors



Experimental Study on Deep Web

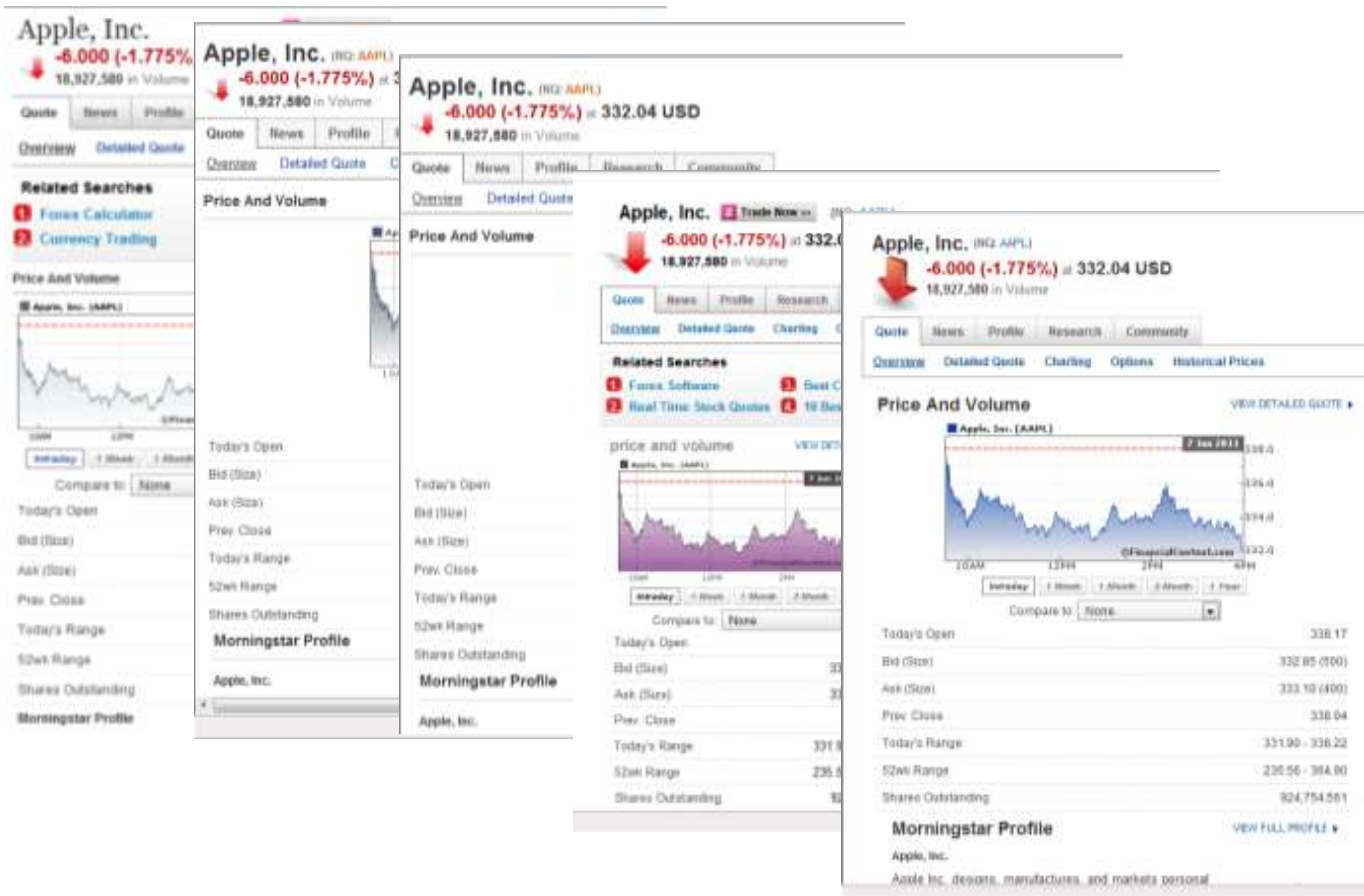
- Why such inconsistency?
  - Pure errors

FlightView **FlightAware** **Orbitz**



Experimental Study on Deep Web

♦ Copying between sources?



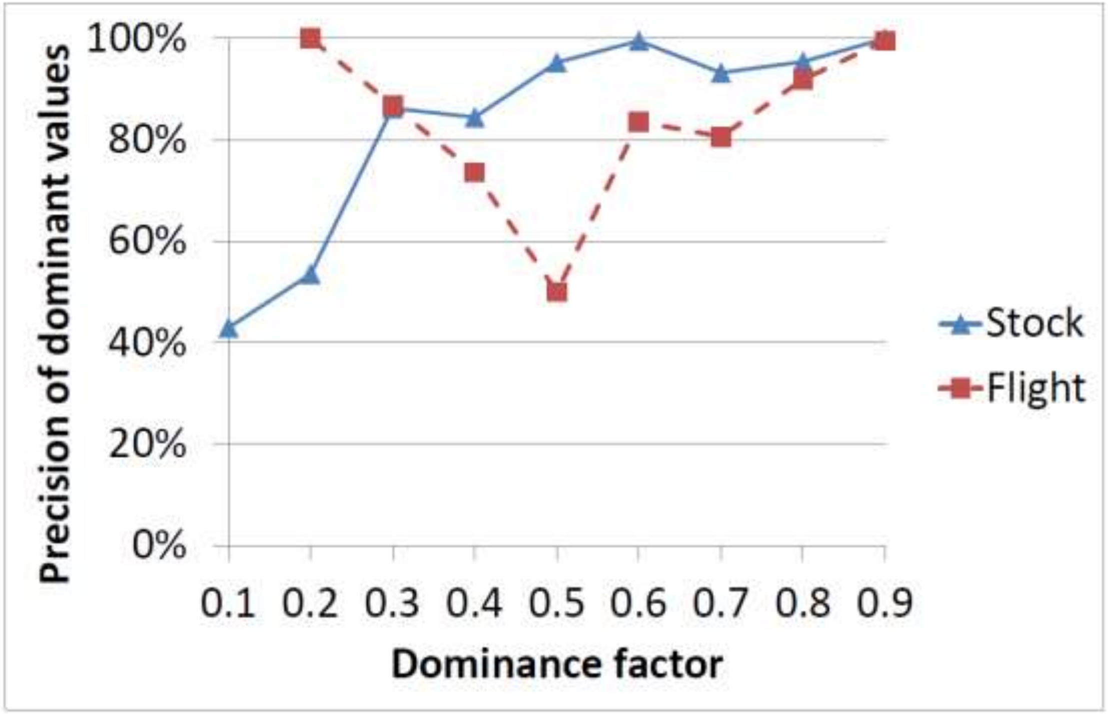
Experimental Study on Deep Web

- Copying on erroneous data?

	Remarks	Size	Schema sim	Object sim	Value sim	Avg accu
Stock	Depen claimed	11	1	.99	.99	.92
	Depen claimed	2	1	1	.99	.75
Flight	Depen claimed	5	0.80	1	1	.71
	Query redirection	4	0.83	1	1	.53
	Dependence claimed	3	1	1	1	.92
	Embedded interface	2	1	1	1	.93
	Embedded interface	2	1	1	1	.61

Experimental Study on Deep Web

- Basic solution: naïve voting
  - .908 voting precision for Stock, .864 voting precision for Flight
  - Only 70% correct values are provided by over half of the sources



Source Accuracy [DBS09a]

- Computing source accuracy: $A(S) = Avg_{v_i(D) \in S} Pr(v_i(D) \text{ true} \mid \Phi)$
  - $v_i(D) \in S$: S provides value v_i on data item D
  - Φ: observations on all data items by sources S
  - $Pr(v_i(D) \text{ true} \mid \Phi)$: probability of $v_i(D)$ being true

- How to compute $Pr(v_i(D) \text{ true} \mid \Phi)$?

Source Accuracy

- Input: data item D, val(D) = $\{v_0, v_1, \ldots, v_n\}$, Φ
- Output: $Pr(v_i(D) \text{ true} \mid \Phi)$, for i=0,…, n (sum=1)
- Based on Bayes Rule, need $Pr(\Phi \mid v_i(D) \text{ true})$
  - Under independence, need $Pr(\Phi_D(S) \mid v_i(D) \text{ true})$
  - If S provides v_i : $Pr(\Phi_D(S) \mid v_i(D) \text{ true}) = A(S)$
  - If S does not : $Pr(\Phi_D(S) \mid v_i(D) \text{ true}) = (1-A(S))/n$
- Challenge:
  - Inter-dependence between source accuracy and value probability?

Source Accuracy

- Continue until source accuracy converges

Source Accuracy

$$A(S) = \underset{v(D)\in S}{Avg} \Pr(v(D) \mid \Phi)$$

Source Vote Count

$$A'(S) = \ln \frac{nA(S)}{1 - A(S)}$$

ValueVote Count

$$C(v(D)) = \sum_{S \in \bar{S}(v(D))} A'(S)$$

Value Probability

$$\Pr(v(D) \mid \Phi) = \frac{e^{C(v(D))}}{\sum_{v_0 \in val(D)} e^{C(v_0(D))}}$$

Value Similarity

- Continue until source accuracy converges

Source Accuracy

$$A(S) = \underset{v(D)\in S}{Avg} \Pr(v(D) \mid \Phi)$$

Source Vote Count

$$A'(S) = \ln \frac{nA(S)}{1 - A(S)}$$

ValueVote Count

$$C(v(D)) = \sum_{S \in \bar{S}(v(D))} A'(S)$$

Value Probability

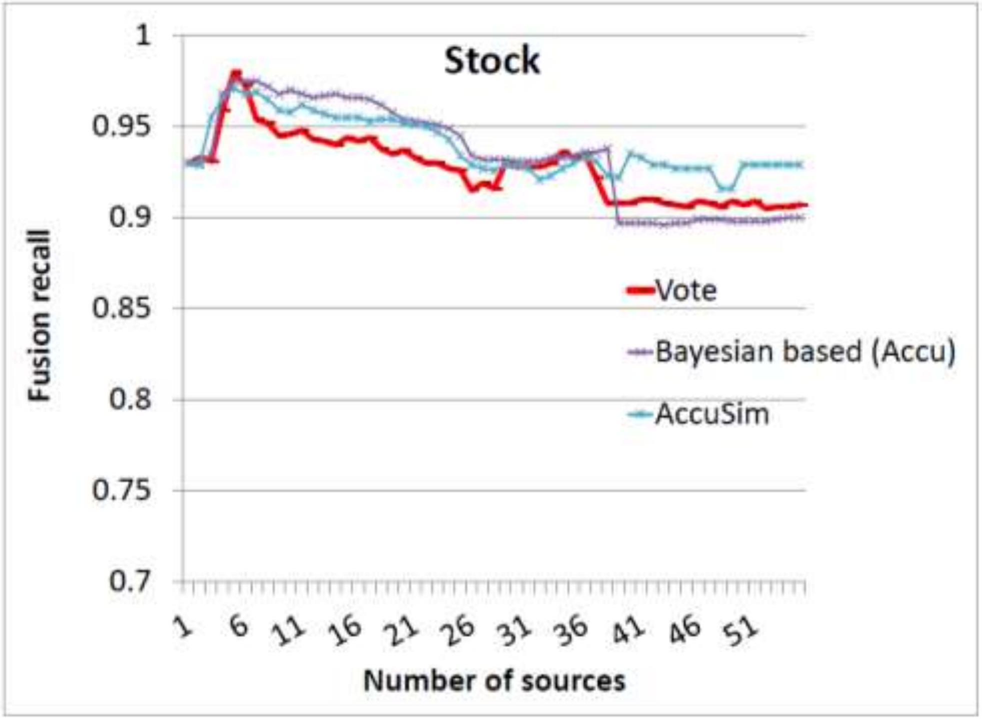
$$\Pr(v(D) \mid \Phi) = \frac{e^{C(v(D))}}{\sum_{v_0 \in val(D)} e^{C(v_0(D))}}$$

- **Consider value similarity**

$$C^*(v) = C(v) + \rho \sum_{v' \neq v} C(v') \bullet sim(v, v')$$

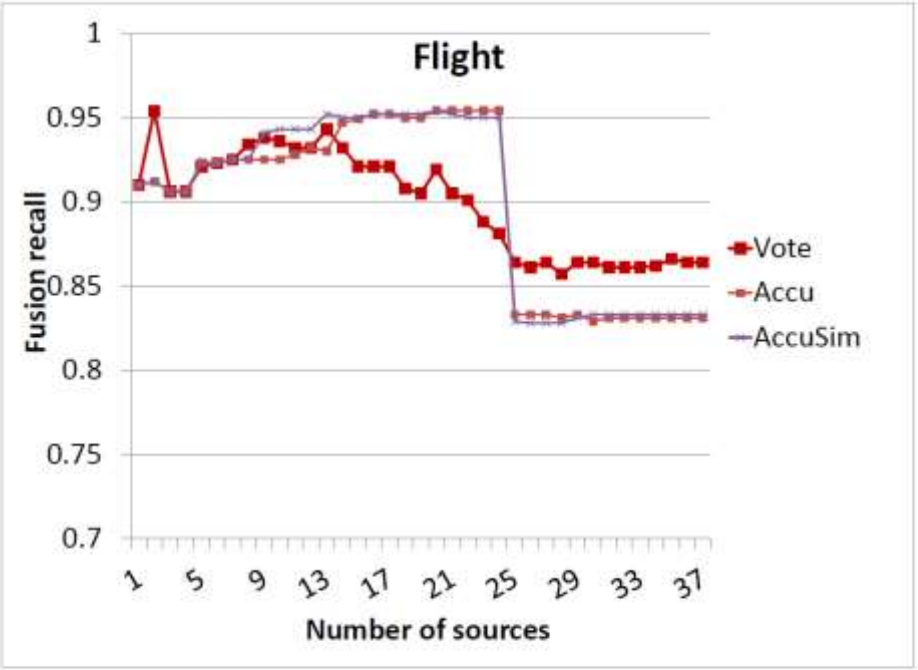
Experimental Study on Deep Web

- Result on Stock data
  - AccuSim's final precision is .929, higher than other methods



Experimental Study on Deep Web

- Result on Flight data
  - AccuSim's final precision is .833, lower than Vote (.857); why?



Experimental Study on Deep Web

- Copying on erroneous data

	Remarks	Size	Schema sim	Object sim	Value sim	Avg accu
Stock	Depen claimed	11	1	.99	.99	.92
	Depen claimed	2	1	1	.99	.75
Flight	Depen claimed	5	0.80	1	1	.71
	Query redirection	4	0.83	1	1	.53
	Dependence claimed	3	1	1	1	.92
	Embedded interface	2	1	1	1	.93
	Embedded interface	2	1	1	1	.61

Copy Detection

Are Source 1 and Source 2 dependent? **Not necessarily**

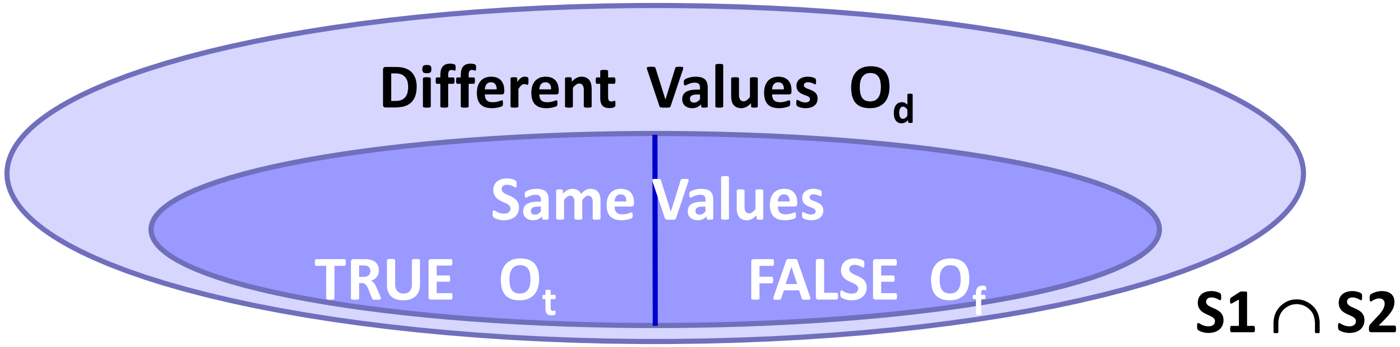
Source 1 on USA Presidents:	Source 2 on USA Presidents:
1st : George Washington	1st : George Washington
2nd : John Adams	2nd : John Adams
3rd : Thomas Jefferson	3rd : Thomas Jefferson
4th : James Madison	4th : James Madison
…	…
41st : George H.W. Bush	41st : George H.W. Bush
42nd : William J. Clinton	42nd : William J. Clinton
43rd : George W. Bush	43rd : George W. Bush
44th: Barack Obama	44th: Barack Obama

Copy Detection

Are Source 1 and Source 2 dependent? Very likely

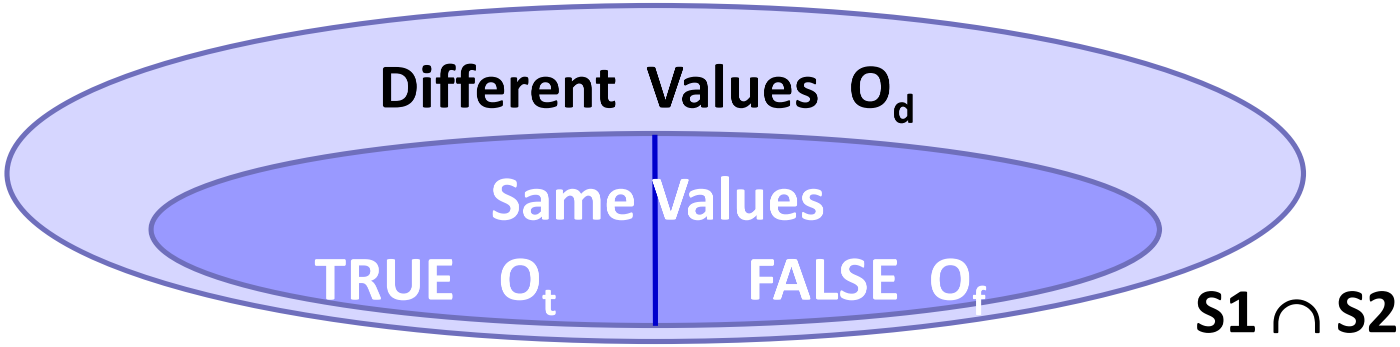
Source 1 on USA Presidents:	Source 2 on USA Presidents:	
1st : George Washington	1st : George Washington	✓
2nd : Benjamin Franklin	2nd : Benjamin Franklin	✘
3rd : John F. Kennedy	3rd : John F. Kennedy	✘
4th : Abraham Lincoln	4th : Abraham Lincoln	✘
…	…	
41st : George W. Bush	41st : George W. Bush	✘
42nd : Hillary Clinton	42nd : Hillary Clinton	✘
43rd : Dick Cheney	43rd : Dick Cheney	✘
44th: Barack Obama	44th: John McCain	

Copy Detection: Bayesian Analysis



- Goal: $Pr(S1 \perp S2 | \Phi)$, $Pr(S1 \sim S2 | \Phi)$ (sum = 1)
- According to Bayes Rule, we need $Pr(\Phi | S1 \perp S2)$, $Pr(\Phi | S1 \sim S2)$
- Key: compute $Pr(\Phi_D | S1 \perp S2)$, $Pr(\Phi_D | S1 \sim S2)$, for each $D \in S1 \cap S2$

Copy Detection: Bayesian Analysis



Pr	Independence	Copying
O_t	A^2	$<$ $A \cdot c + A^2(1-c)$
O_f	$\frac{(1-A)^2}{n}$	$\ll$ $(1-A) \bullet c + \frac{(1-A)^2}{n}(1-c)$
O_d	$P_d = 1 - A^2 - \frac{(1-A)^2}{n}$	$>$ $P_d(1-c)$

Discount Copied Values

- Continue until convergence

Source Accuracy

$$A(S) = \underset{v(D)\in S}{Avg} \Pr(v(D) \mid \Phi)$$

Source Vote Count

$$A'(S) = \ln \frac{nA(S)}{1 - A(S)}$$

ValueVote Count

$$C(v(D)) = \sum_{S \in \bar{S}(v(D))} A'(S)$$

Value Probability

$$\Pr(v(D) \mid \Phi) = \frac{e^{C(v(D))}}{\sum_{v_0 \in val(D)} e^{C(v_0(D))}}$$

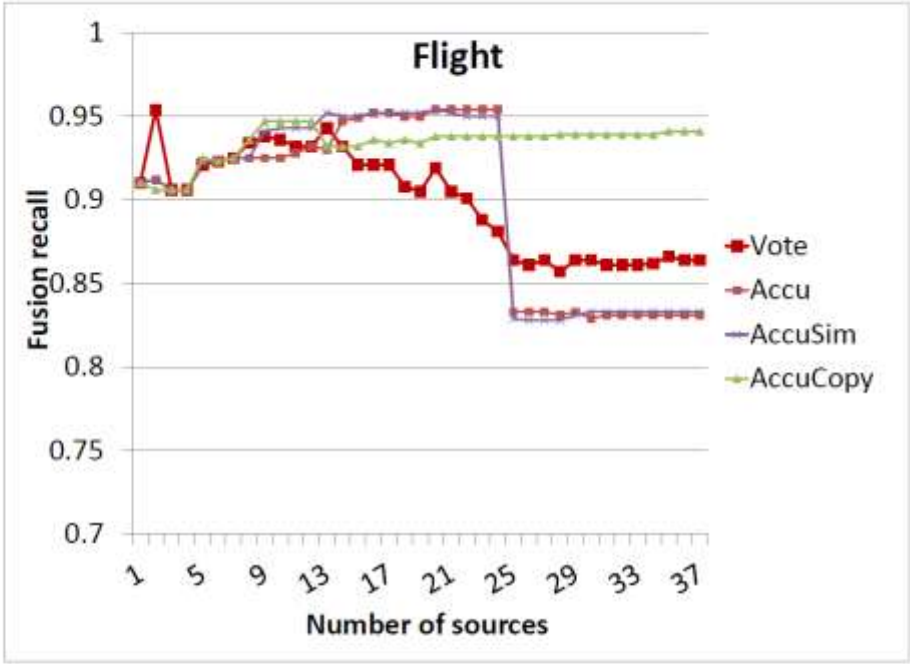
- **Consider dependence**

$$C(v) = \sum_{S \in \bar{S}(v)} A'(S) \bullet I(S)$$

- **I(S)- Pr of independently providing value v**

Experimental Study on Deep Web

- Result on Flight data
  - AccuCopy's final precision is .943, much higher than Vote (.864)



Summary

	Schema alignment	Record linkage	Data fusion
Volume	• Integrating deep Web • Web table/lists	• Adaptive blocking	• Online fusion
Velocity	• Keyword-based integration for dynamic data	• Incremental linkage	• Fusion for dynamic data
Variety	• Dataspaces • Keyword-based integration	• Linking texts to structured data	• Combining fusion with linkage
Veracity		• Value-variety tolerant RL	• Truth discovery

Outline

- Motivation
- Schema alignment
- Record linkage
- Data fusion
- Future work

Future Work

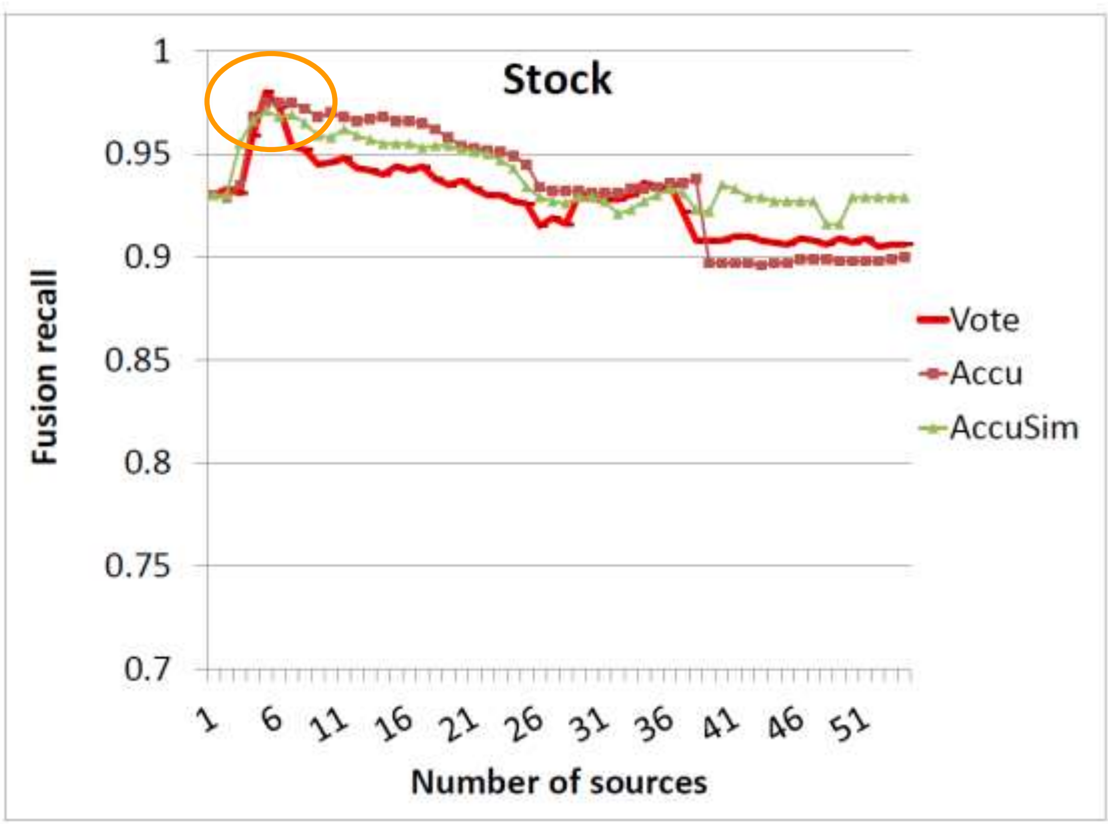
- Reconsider the architecture

Data warehousing

Virtual integration

Future Work

- The more, the better?



Future Work

- Combining different components