DATA FUSION –
a stage in data integration

For the Course on Information Extraction and Integration from Big Data

Lectures on Data Fusion: an Outline

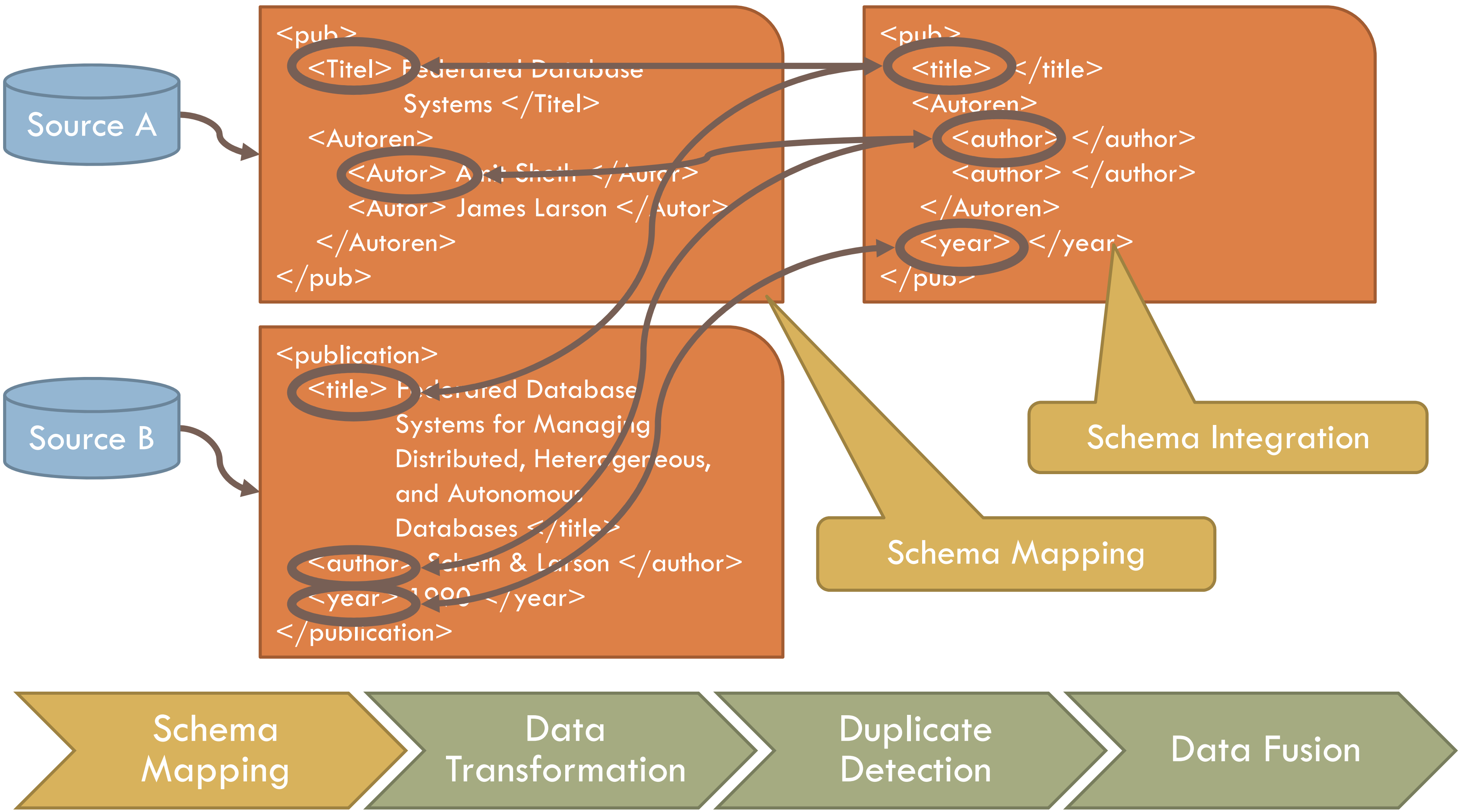
1st Lecture

1. Data Integration – an Introduction
2. Data Fusion as a stage of Database Integration
3. Conflict resolution strategies and functions at the fusion stage
4. Data fusion and conflict resolution operators
5. Data Fusion: Join-based Approaches
6. Data Fusion: Union-based Approaches

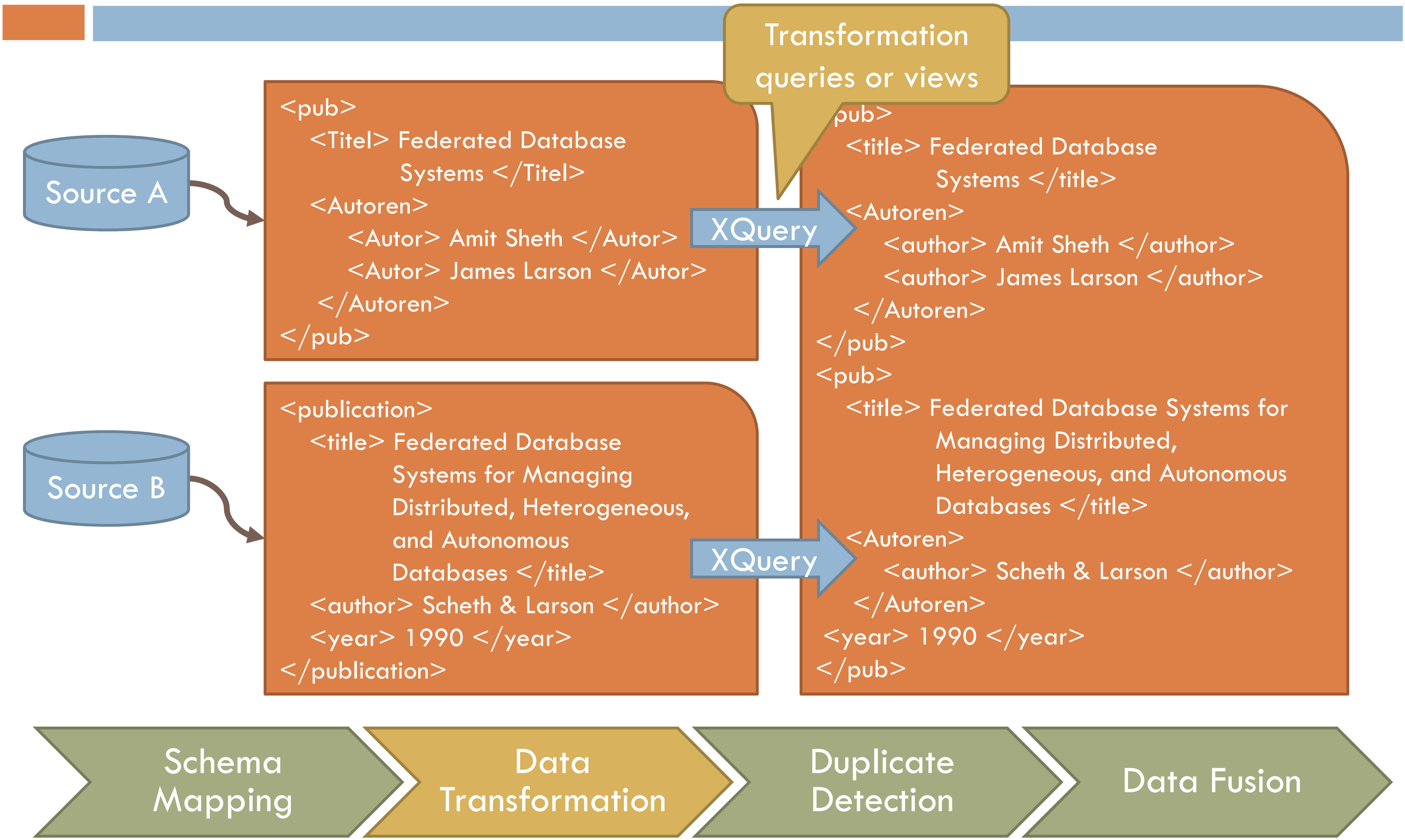
2nd Lecture

1. Data Fusion Operator
2. Illustration of various data fusion operators use
3. Another approach for data fusion
4. Data fusion in existing integration systems
5. Truth-discovery techniques
6. Open Problems and Challenges
7. References

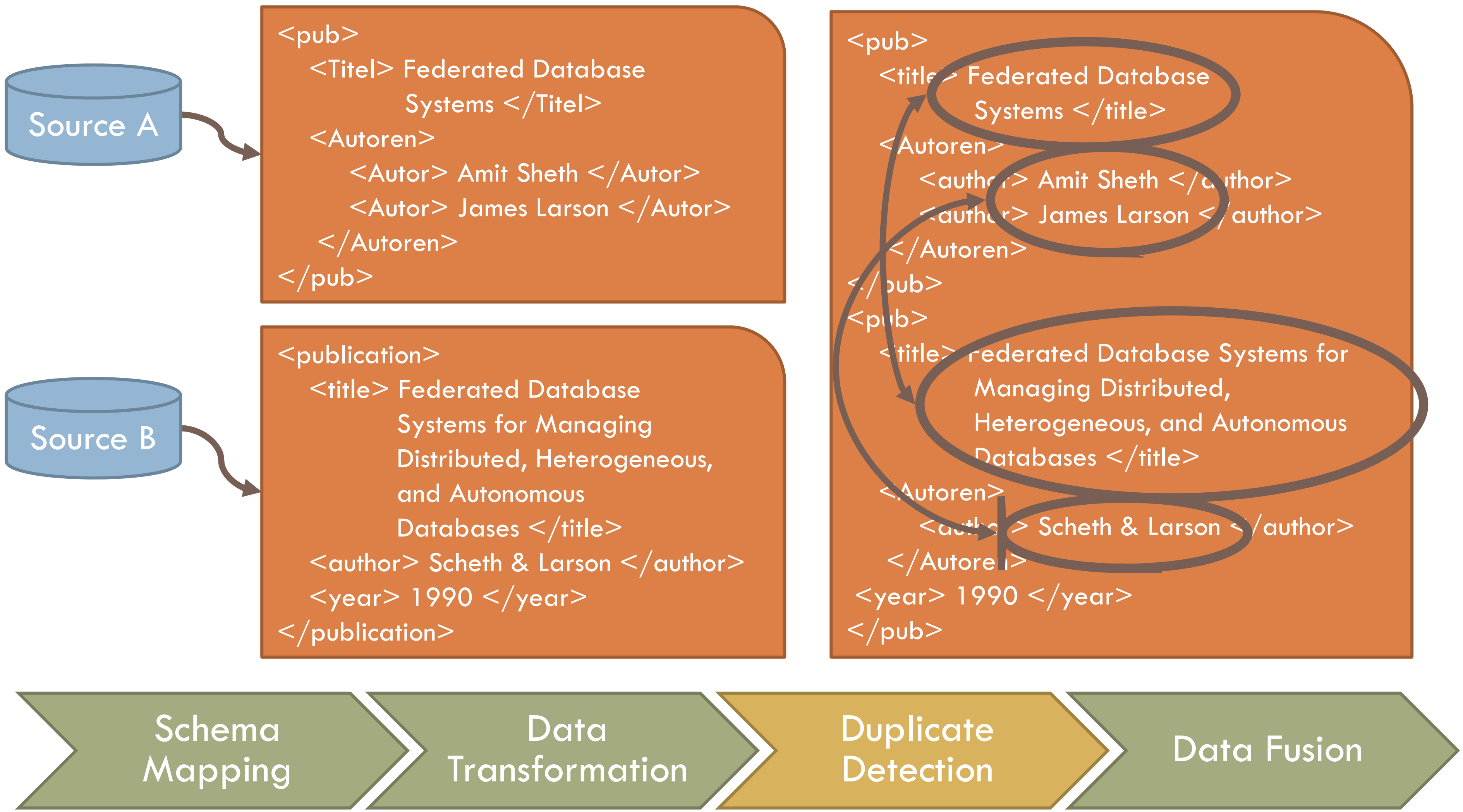
Information Integration



Information Integration



Information Integration



Information Integration

Source A

```
<pub>
   <Titel> Federated Database
            Systems </Titel>
   <Autoren>
       <Autor> Amit Sheth </Autor>
       <Autor> James Larson </Autor>
    </Autoren>
</pub>
```

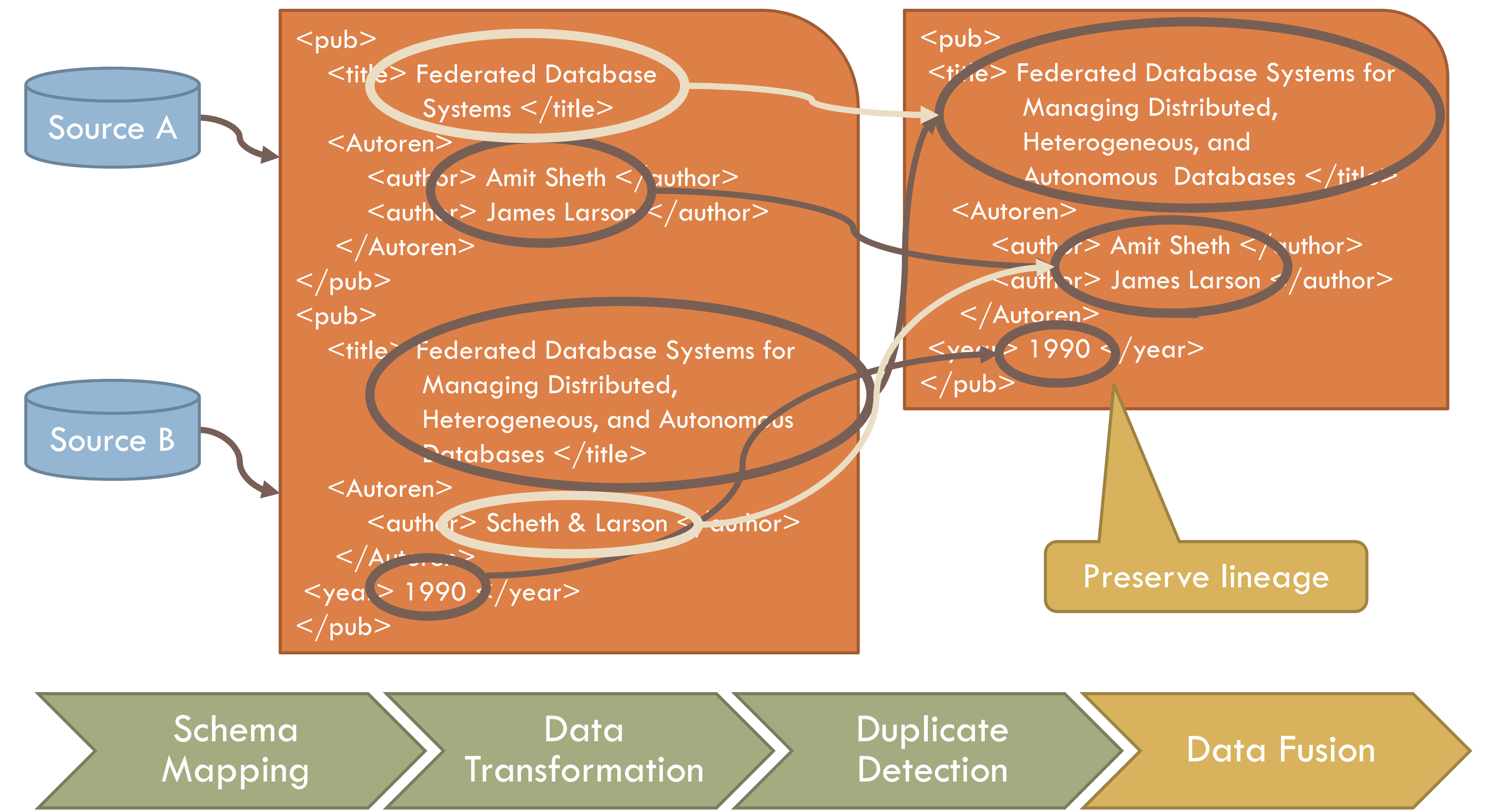
Source B

```
<publication>
   <title> Federated Database
           Systems for Managing
           Distributed, Heterogeneous,
           and Autonomous
           Databases </title>
   <author> Scheth & Larson </author>
   <year> 1990 </year>
</publication>
```

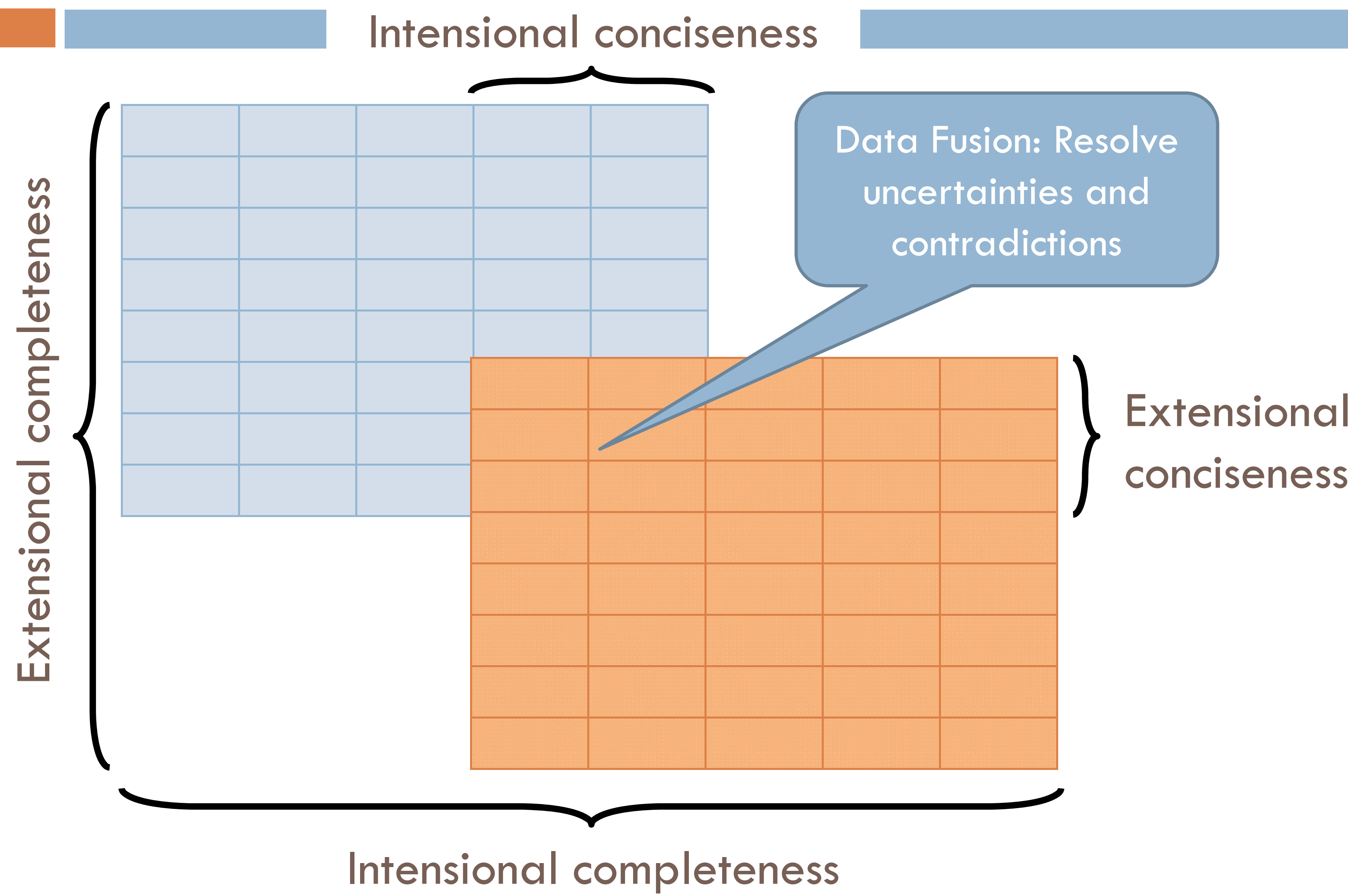
```
<pub>
   <title> Federated Database
            Systems </title>
   <Autoren>
       <author> Amit Sheth </author>
       <author> James Larson </author>
    </Autoren>
</pub>
<pub>
   <title> Federated Database Systems for
            Managing Distributed,
            Heterogeneous, and Autonomous
            Databases </title>
   <Autoren>
       <author> Scheth & Larson </author>
    </Autoren>
 <year> 1990 </year>
</pub>
```

Schema Mapping → Data Transformation → Duplicate Detection → Data Fusion

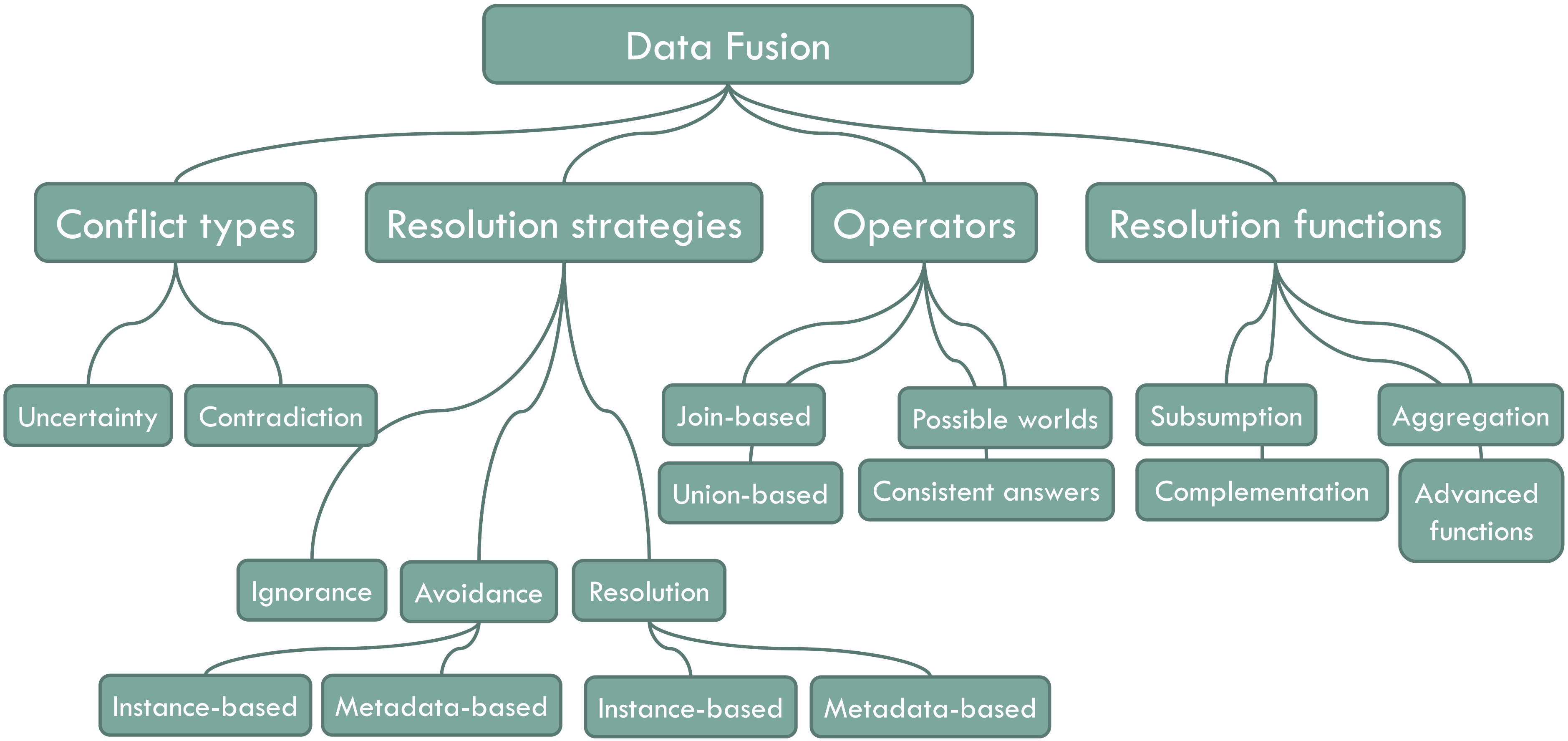
Information Integration



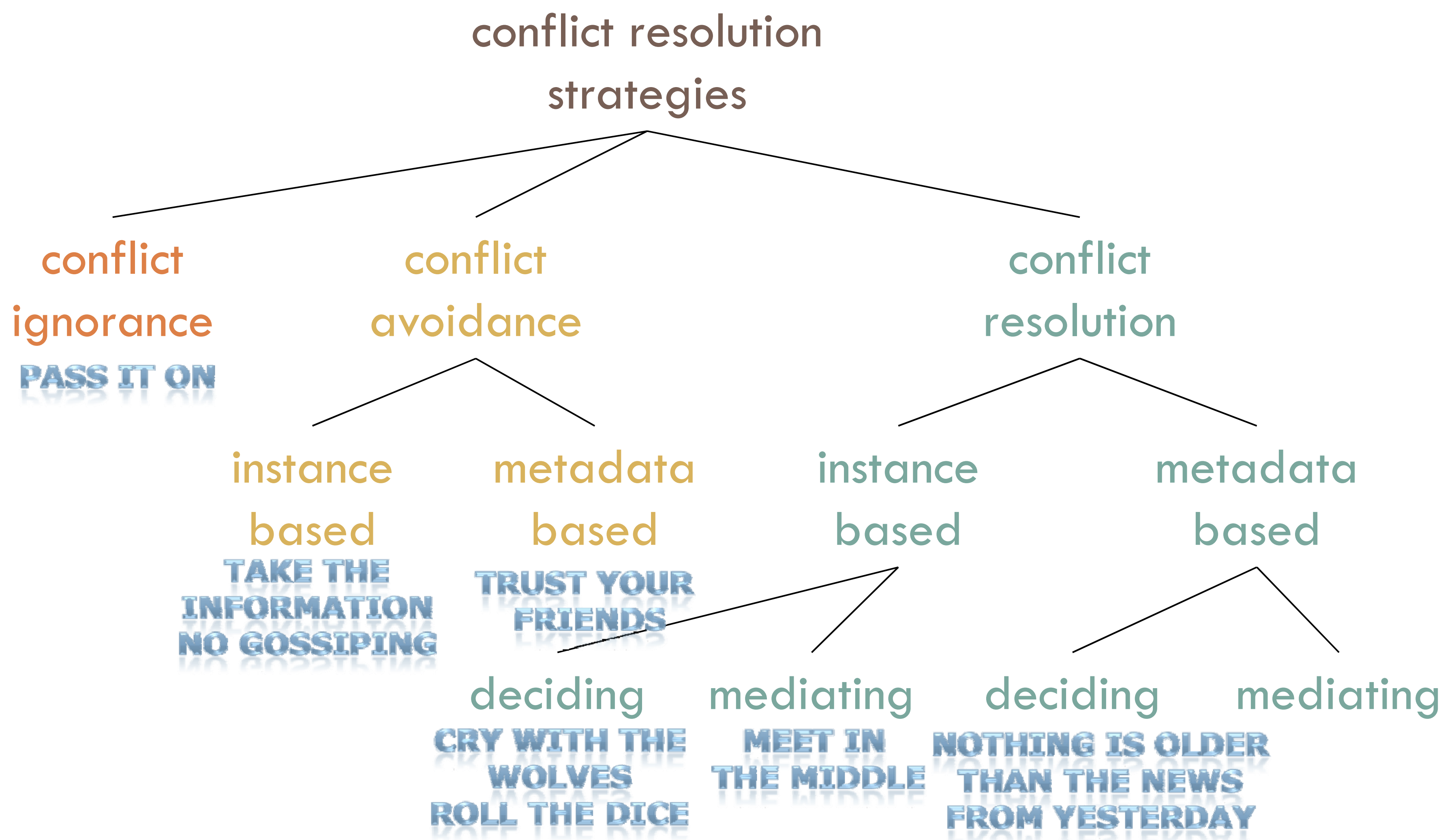
Completeness, Conciseness, and Correctness



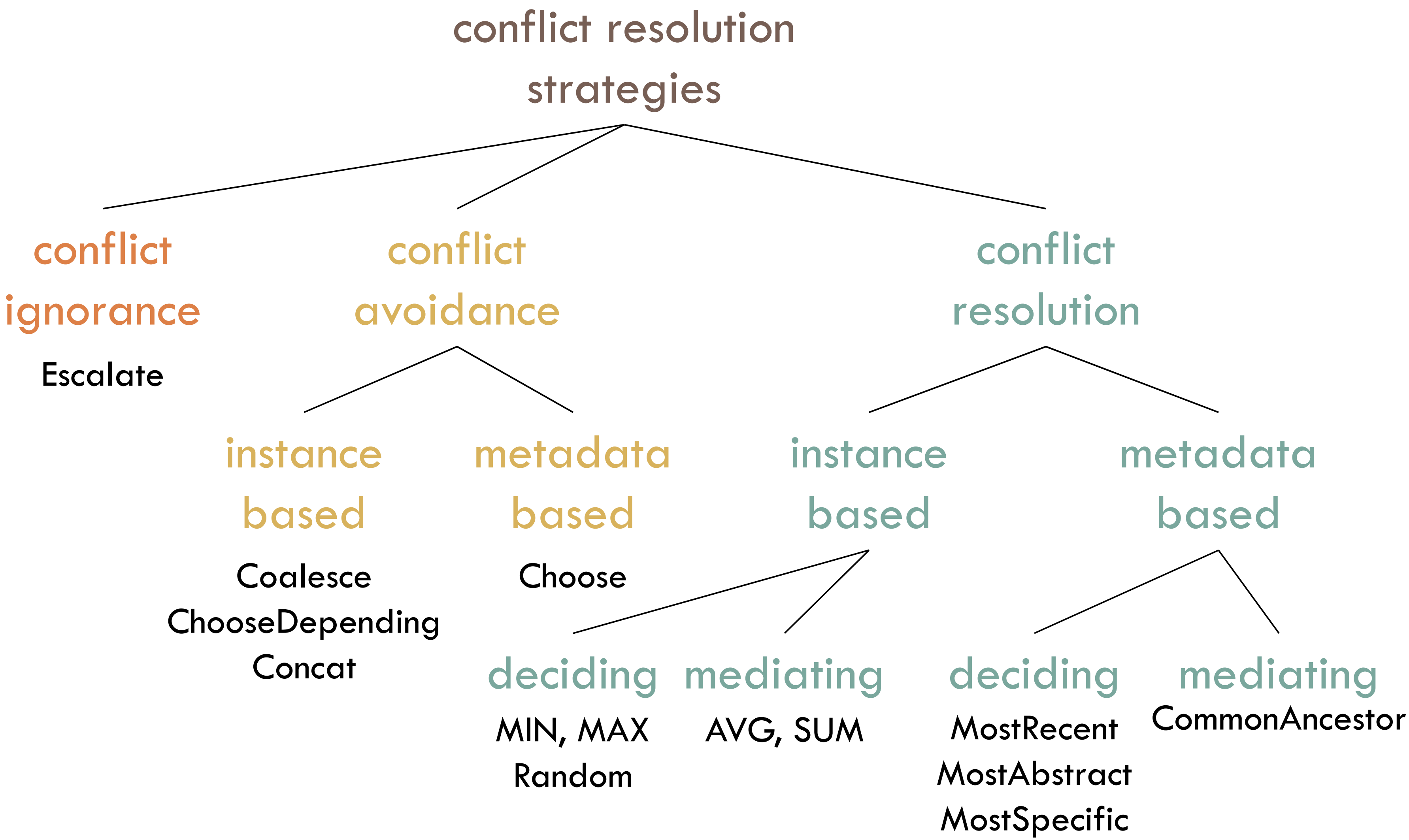
The Field of Data Fusion



Classification of Strategies



Classification of Functions



Conflict Resolution Functions

Function	Description	Examples
Min, Max, Sum, Count, Avg	Standard aggregation	NumChildren, Salary, Height
Random	Random choice	Shoe size
Longest, Shortest	Longest/shortest value	First_name
Choose(source)	Value from a particular source	DoB (DMV), CEO (SEC)
ChooseDepending(val, col)	Value depends on value chosen in other column	city & zip, e-mail & employer
Vote	Majority decision	Rating
Coalesce	First non-null value	First_name
Group, Concat	Group or concatenate all values	Book_reviews
MostRecent	Most recent (up-to-date) value	Address
MostAbstract, MostSpecific, CommonAncestor	Use a taxonomy / ontology	Location
Escalate	Export conflicting values	gender
…	…	…

Minimum Union

- Union: Elimination of exact duplicates
- Minimum Union: Elimination of subsumed tuples
  - Outer union
  - Subsumption
- Rewriting in SQL using DWH extensions (Windows) and assuming existence of favorable ordering [RPZ04]

A	B	C
a	b	c
e	f	g
m	n	o

$\uplus$

A	B	D
a	b	$\perp$
e	f	h
m	p	$\perp$

=

A	B	C	D
a	b	c	$\perp$
a	b	$\perp$	$\perp$
e	f	g	$\perp$
e	f	$\perp$	h
m	n	o	$\perp$
m	p	$\perp$	$\perp$

A tuple t1 subsumes a tuple t2, if it has same schema, has less NULL-values, and coincides in all non-NULL-values.

R↓

A	B	C	D
a	b	c	$\perp$
e	f	g	$\perp$
e	f	$\perp$	h
m	n	o	$\perp$
m	p	$\perp$	$\perp$

Complement Union – Proposal

- Elimination of complementing tuples
  - Outer union
  - Complementation
- No known SQL rewriting

<u>A</u>	B	C
a	b	c
e	f	g
m	n	o

$\uplus$

<u>A</u>	B	D
a	b	$\perp$
e	f	h
m	p	$\perp$

=

A	B	C	D
a	b	c	$\perp$
a	b	$\perp$	$\perp$
e	f	g	$\perp$
e	f	$\perp$	h
m	n	o	$\perp$
m	p	$\perp$	$\perp$

R ⇅

A	B	C	D
a	b	c	$\perp$
e	f	g	h
m	n	o	$\perp$
m	p	$\perp$	$\perp$

A tuple t1 complements a tuple t2, if it has same schema and coincides in all non-NULL-values.

Includes duplicate removal and subsumption

Merge and Prioritized Merge

A is real-world ID

A	B	C
a	b	c
e	f	g
m	n	o
m	n	⊥
q	r	s

|⋈

A	B	D
a	b	⊥
e	f	h
m	p	⊥

=

A	B	C	D
a	COAL(b,b)	c	⊥
e	COAL(f,f)	g	h
m	COAL(n,p)	o	⊥
m	COAL(n,p)	⊥	⊥
q	r	s	⊥

=

A	B	C	D
a	b	c	⊥
e	f	g	h
m	n	o	⊥
m	n	⊥	⊥
q	r	s	⊥

∪

A	B	C
a	b	c
e	f	g
m	n	o
m	n	⊥
q	r	s

⋈|

A	B	D
a	b	⊥
e	f	h
m	p	⊥

=

A	B	C	D
a	COAL(b,b)	c	⊥
e	COAL(f,f)	g	h
m	COAL(p,n)	o	⊥
m	COAL(p,n)	⊥	⊥

=

A	B	C	D
~~a~~	~~b~~	~~c~~	~~⊥~~
~~e~~	~~f~~	~~g~~	~~h~~
m	p	o	⊥
m	p	⊥	⊥

Data Fusion: Join-based Approaches

Join approaches for fusion

- **Join approaches**: join approaches generally retain information on same real-world objects in different columns (not in different rows). Columns with equal semantic attributes do not automatically coincide as in the union approaches. They need to be combined explicitly.
- **Completeness comes more or less without cost; but conciseness needs to be achieved by an additional processing of the join result**.
- If there are no intrasource duplicates in the tables, a simple combination of two rows using a user-defined function is sufficient. However, **if there are intrasource duplicates grouping and aggregation or some advanced join operator is needed.**

Join-based operators (1)

1) ***1) Standard Joins*****.** An **equi-join** combines tuples from two relations if the join condition consists of equality conditions between columns. The **key-join** is uniqueness preserving but not necessarily value/object preserving. Same is true for the **natural join**. The **full outer join** extends the result of a standard join operation by adding tuples that are only included in one of the source relations.

2) ***2) Full Disjunction*** The full disjunction operator is defined [1994] as the combination of two or more tables, where all matching tuples are combined into one single tuple. Full disjunctions are **uniqueness preserving** and **value/object preserving**, like full outer joins. Full disjunction can generally can not be computed by a combination of outer joins alone [Galindo-Legaria 1994].

Join-based operators (2)

3) Match Join and Conflict Tolerant Queries. In a first step, corresponding attribute values from all sources are **separately projected out and combined by union**, expanded by their corresponding real-world identifier. Then, the **real-world identifiers are used to rejoin**, resulting in one single large table. The operation to create this table is called **match join**. In a second step, to increase conciseness, **tuples are selected from this table according to the accompanying conflict-tolerant query model**. When selecting tuples, **it uses a selection predicate and observes the evaluation parameter** of the operation. This parameter might be:

high confidence: the selection predicate needs to be true for *all* tuples for each real-world identifier;

random evidence : verifies the predicate for a random tuple per real-world identifier;

possible at all : the selection predicate needs to be true for at least one tuple for each real-world identifier.

In the end there remains only one representation per real-world identifier, thus the operator is **uniqueness enforcing**, creating a **conflict-free** result, but, depending on the parameterization, **value/object preserving**.

Full Disjunction

- Represents all possible combinations of source tuples
  - Full outer join on all common attributes
    - All combinations for more than two sources
  - Minimum union over results (redundant tuples removing)
- Combines complementing tuples (only inter-source)
- Algorithms: [GL94,RU96,CS05]

<u>A</u>	B	C
a	b	c
e	f	g
k	⊥	o
k	m	⊥

$|\bowtie|$

<u>A</u>	B	D
a	b	⊥
e	f	h
m	p	⊥
k	q	r

=

A	B	C	D
a	b	c	⊥
e	f	g	h
m	p	⊥	⊥
k	⊥	o	⊥
k	m	⊥	⊥
k	q	⊥	r

R ↓

A	B	C	D
a	b	c	⊥
e	f	g	h
m	p	⊥	⊥
k	⊥	o	⊥
k	m	⊥	⊥
k	q	⊥	r

Relational Operators: Full Disjunction

Name	Age	Country
Adam	20	England
Ben	23	USA
Chris	NULL	Canada
Chris	21	NULL

$|\bowtie|$

Name	Age	Height
Adam	20	NULL
Ben	23	174
David	40	NULL
Chris	33	183

=

Name	Age	Country	Height
Adam	20	England	NULL
Ben	23	USA	174
David	40	NULL	NULL
Chris	NULL	Canada	NULL
Chris	21	NULL	NULL
Chris	33	NULL	183

$\xrightarrow[\text{min un}]{}$

Name	Age	Country	Height
Adam	20	England	NULL
Ben	23	USA	174
David	40	NULL	NULL
Chris	NULL	Canada	NULL
Chris	21	NULL	NULL
Chris	33	NULL	183

Match Join

- Context: AURORA Project [YÖ99]
- Handles columns individually using projections (with IDs)
- Performs UNION on each column across all sources
- Reassembles using FULL OUTER JOINS
- Uses "conflict-tolerant query model" to query these possible worlds.

```
WITH OU(A,B,C,D) AS (
  ( SELECT A, B, C, NULL AS D FROM U1 )
    UNION
  ( SELECT  A, B, NULL AS C, D FROM U2 ) ),
  B_V (A,B) AS ( SELECT DISTINCT  A, B FROM OU ),
  C_V (A,C) AS ( SELECT DISTINCT A, C FROM OU ),
  D_V (A,D) AS ( SELECT DISTINCT A, D FROM OU ),
SELECT A, B, C, D
FROM B_V FULL OUTER JOIN C_V FULL OUTER JOIN D_V
ON B_V.A=C_V.A AND C_V.A=D_V.A
```

Match Join

- Conflict-tolerant query model
  - Chooses tuples from result of MatchJoin
- Three semantics
  - HighConfidence, RandomEvidence, PossibleAtAll
- Resolution functions
  - SUM, AVG, MAX, MIN, ANY, DISCARD

```
SELECT  ID, Name[ANY], Age[MAX]
FROM    MatchJoin(U1,U2)
WHERE   Age>22
WITH    PossibleAtAll
```

Grouping and Aggregation

- Outer union then group by real-world ID
- Aggregate all other columns using conflict resolving aggregate function
- Efficient implementations
- Catches inter- and intra-source duplicates
- Restricted to built-in aggregate-functions
  - MAX, MIN, AVG, VAR, STDDEV, SUM, COUNT

```
WITH OU AS (
  ( SELECT A, B, C, NULL AS D FROM U1 )
      UNION (ALL)
  ( SELECT A, B, NULL AS C, D FROM U2 ) ),

SELECT   A, MAX(B),  MIN(C), SUM(D)
FROM     OU
GROUP BY A
```

Relational Operators: Grouping And Aggregation

- Combination of
  - Outer union
  - Group by (real-world) ID
  - Aggregation of all other columns using conflict resolving aggregate functions (AVG, SUM, MIN, MAX, COUNT, …)

Relational Operators: Grouping And Aggregation

```
WITH OuterUnionAS (
( SELECT Name, Age, Country, NULL AS Height
FROM R )
UNION
( SELECT Name, Age, NULL AS Country, Height
FROM S) ),

SELECT Name, AVG(Age), COUNT(Country), MAX(Height)
FROM OuterUnion
GROUP BY Name
```

Data Fusion Operator

Conflict resolution by Data Fusion operator

- **Conflict resolution is a part of the *data fusion* operator.**
- Conflict resolution (CR) on a tuple set *T* can be seen as combining a *sort* on that tuple set *T* and an order observing *aggregation*, where not only the standard aggregation functions are allowed. The *result of* conflict resolution *is **one single tuple t** that complies to an output schema OS. Conflicts are resolved attribute-wise by a specific conflict resolution functions. Given the ordered set of tuples T, for each attribute a_i an ordered set of values V_i is defined. The result of* conflict resolution *is then the result of applying all conflict resolution functions in CR on the sets V_i.*
- In case of functions with additional input (metadata conflict resolution functions), additional parameters to the conflict resolution functions are first passed to the operator and then to the respective functions.

Data fusion operators

- **The *data fusion* operator goes beyond *subsumption* and *minimum union* and is also able to resolve inconsistencies.**
- **Data Fusion Operator without Removal of Subsumed Tuples**. In the schema of input relation two disjoint sets of attributes are included: the identifying objects attributes (F) and the set of conflicting attributes (CA). The operator uses as parameters a set of conflict resolution functions to create (attribute-wise) values of the respective attributes (corresponding to the conflicting ones) of the output relation.
- **Data Fusion Operator with Removal of Subsumed Tuples**: additionally to the above subsumed tuples are removed.
- **Semantics of data fusion**: 1) the source relation is divided into groups of tuples using the attributes in *F* to identify groups; 2) for the case with subsumed tuple removal such tuples are removed from each group; 3) then a conflict resolution is executed on each group separately.

Conflict resolution operator

Definition (Conflict Resolution Operator)

The conflict resolution operator is denoted $\lambda_{CR,O}(T)$ and is defined as the application of a set of m conflict resolution functions $CR = \{cr_1, cr_2, \ldots, cr_m\}$ on an ordered set of n tuples $T = \{t_1, t_2, \ldots, t_n\}$ that comply to an input schema $IS = (name^I, A^I, dom^I)$ with l attributes $A^I = \{a_1^I, \ldots, a_l^I\}$. The result of conflict resolution is one single tuple t that complies to an output schema $OS = (name^O, A^O, dom^O)$, with m attributes $A^O \subseteq A^I$.

Conflict resolution on a tuple set T can be seen as combining a *sort* on that tuple set T and an order observing *aggregation*, where not only the standard aggregation functions are allowed

In case of functions with additional input (metadata conflict resolution functions), additional parameters to the conflict resolution functions are first passed to the operator and then to the respective functions.

Two variants of the *data fusion* operator: 1) Without removal of subsumed tuples; 2) With removal of subsumed tuples

Two phases of Data Fusion operator

- The *data fusion* operator is defined as a unary operator. **So, in order to fuse data from multiple source relations, all tuples from all source relations involved need to be combined prior to that, in order to form a single relation**. Here, for data integration, an *outer union* can be used, as well as a *join* or any other suitable operator.
- **Step 1: Preparing for conflict resolution.** First, all tuples that describe the same real-world object are grouped together. This is done by doing a grouping on the column(s) given in the list of identifying attributes F = { f1, . . . , fk}. Duplicate detection needs to be done prior to the fusion process.
- **Step 2: Increasing conciseness and resolving conflicts.** Then, in case of the second variant of the data fusion operator, subsumed tuples are removed per group. In case of the first variant, subsumed tuples are kept. Interestingly, removing subsumed tuples per group as needed in our case does not yield the same result as removing subsumed tuples from the entire table. All remaining tuples of one group are then fused together into only one single tuple, at the same time resolving inconsistencies and data conflicts. This is done by applying *conflict resolution* on the tuple set by conflict resolution operator that is formed.

FUSE BY

- SQL extensions to resolve uncertainties and contradictions [BN05,BBB+05]
- FUSE FROM implies OUTER UNION
  - Removes subsumed and duplicate tuples by default
- FUSE BY declares real-world ID
- RESOLVE specifies conflict resolution function from catalog
  - Default: COALESCE
- Implemented on top of relational DBMS “XXL”

```
SELECT    ID,
          RESOLVE(Title, Choose(IMDB)),
          RESOLVE(Year, Max), RESOLVE(Director),
          RESOLVE(Rating), RESOLVE(Genre, Concat)
FUSE FROM IMDB, Filmdienst
FUSE BY (ID)
ON ORDER Year DESC
```

Illustration of various data fusion operators use

Summary of Operators

	Duplicates	Subsumed tuples	Complementing tuples	Contradictions
Union, Outer Union	✓	✗	✗	✗
Minimum Union	✓	✓	✗	✗
Full Disjunction	✓	✓	✓ (inter-source)	
Complement Union	✓	✓	✓	✗
Merge	✓	✓ (inter-source)	✓ (inter-source)	✗
MatchJoin + CTQM	✓	✓	✓	✗ ✓
Group By	✓	✓	✓	✓
Fuse By	✓	✓	✓	✓

Data sources for the illustration (Police table)

Police table (1.1) contains information about missing persons and the Hospital table (1.2) contains information about persons admitted to a hospital. We want to find out if a missed person has already been found. We need to fuse both table as *concise* and as *complete* as possible.

Relation *Police*

	Name	Birthdate	Sex	Address
1	Miller	7/7/1959	m	234 Main St.
2	Miller	⊥	⊥	234 Main St.
3	Peters	1/19/1953	m	43 First St.
8	Smith	8/9/1970	m	Mass Ave.

Table 1.1: Information about missing persons as given by the police.

Data sources for the illustration (Hospital table)

Relation *Hospital*

	Name	Birthdate	Sex	Blood
4	Peters	1/19/1953	$\perp$	AB
5	Peters	1/19/1953	m	$\perp$
6	Miller	$\perp$	f	B
7	Miller	7/7/1959	m	0
9	Smith	9/8/1970	m	A

Table 1.2: Information about persons admitted to a hospital.

Outer join and conflict resolution (Name for ID)

Conflicts were removed as follows: Miller “m” for the Sex, Birthday for Smith as “8/9/1970”. Under such assumption the result is concise and complete. Though of course there is a lost of information (Mrs. Miller has been lost. Combination of Name and Sex should be considered as ID (such choice will be used in the remaining examples)

Data Fusion of *Police* and *Hospital*

Name	Birthdate	Sex	Address	Blood
Miller	7/7/1959	m	234 Main St.	0
Peters	1/19/1953	m	43 First St.	AB
Smith	8/9/1970	m	Mass Ave.	A

Table 1.3: Combining Table 1.1 and Table 1.2 using *outer union* and conflict resolution functions.

Outer union

All tuples from the sources are included into the result. Although the result is complete, it is not concise (no deduplication actions have been applied)

Outer Union of *Police* and *Hospital*

	Name	Birthdate	Sex	Address	Blood
1	Miller	7/7/1959	m	234 Main St.	⊥
2	Miller	⊥	⊥	234 Main St.	⊥
3	Peters	1/19/1953	m	43 First St.	⊥
4	Peters	1/19/1953	⊥	⊥	AB
5	Peters	1/19/1953	m	⊥	⊥
6	Miller	⊥	f	⊥	B
7	Miller	7/7/1959	m	⊥	0
8	Smith	8/9/1970	m	Mass Ave.	⊥
9	Smith	9/8/1970	m	⊥	A

Table 1.4: Outer Union result of the two example tables.

Join

Only 3 out of four persons from sources are included, though some of them with contradicting data (in the Smith case). Attribute Birthday is included twice.

Join of *Police* and *Hospital*

	Name	Sex	P.Birthdate	H.Birthday	P.Address	H.Blood
$1 \bowtie 7$	Miller	m	7/7/1959	7/7/1959	234 Main St.	0
$3 \bowtie 5$	Peters	m	1/19/1953	1/19/1953	43 First St.	$\perp$
$8 \bowtie 9$	Smith	m	8/9/1970	9/8/1970	mass Ave.	A

Table 1.5: Join result of the two example tables, using Name and Sex as join attributes.

Consistent query answer on the Outer Union (under constraint ***Unique Name, Sex***)

8/9/1970”. Under such assumption the result is concise and complete. Though the result looks consistent (w.r.t. Unique constraint) , it is incomplete as much information is missing in the result

Consistent Query Answer of *Police* and *Hospital*

	Name	Birthdate	Sex	Address	Blood
2	Miller	$\perp$	$\perp$	234 Main St.	$\perp$
4	Peters	1/19/1953	$\perp$	$\perp$	AB
6	Miller	$\perp$	f	$\perp$	B

Table 1.6: Consistent answer on the Outer Union of the two tables from above

Match Join (Name as ID)

Match Join of *Police* and *Hospital*

Name	Birthdate	Sex	Adress	Blood
Miller	7/7/1959	m	234 Main St.	B
Miller	7/7/1959	m	234 Main St.	0
Miller	7/7/1959	f	234 Main St.	B
Miller	7/7/1959	f	234 Main St.	0
Peters	1/19/1953	m	43 First St.	AB
Smiths	8/9/1970	m	Mass Ave.	AB
Smith	9/8/1970	m	Mass Ave.	AB

Table 1.7: Match Join of the two example tables, using NAME as identifier

Merge (Name and Sex as ID)

Merge on *Police* and *Hospital*

Name	Sex	P.Birthdate	H.Birthday	Address	Blood
Peters	⊥	⊥	1/19/1953	⊥	AB
Peters	m	1/19/1953	1/19/1953	43 First St.	⊥
Miller	f	⊥	⊥	⊥	B
Miller	m	7/7/1959	7/7/1959	234 Main St.	O
Miller	⊥	⊥	⊥	234 Main St.	⊥
Smith	m	8/9/1970	9/8/1970	Mass Ave.	A

Table 1.8: Merge on the two tables from above, using NAME and SEX as identifying attributes

Full Disjunction

This and the next result are nearest to optimal. But the case of complementing tuples (e.g. tuples 1 and 7) is not handled in the approach

Full Disjunction of *Police* and *Hospital*

	Name	Sex	P.Birthdate	H.Birthdate	Address	Blood
4	Peters	⊥	⊥	1/19/1953	⊥	AB
6	Miller	f	⊥	⊥	⊥	B
$1 \bowtie 7$	Miller	m	7/7/1959	7/7/1959	234 Main St.	O
$3 \bowtie 5$	Peters	m	1/19/1953	1/19/1953	43 First St.	⊥
$8 \bowtie 9$	Smith	m	8/9/1970	9/8/1970	Mass Ave.	A

Table 1.9: Full Disjunction of Police and Hospital

Minimum union (tuple submission)

Minimum Union of *Police* and *Hospital*

	Name	Birthdate	Sex	Address	Blood
1+2	Miller	7/7/1959	m	234 Main St.	⊥
3+5	Peters	1/19/1953	m	43 First St.	⊥
4	Peters	1/19/1953	⊥	⊥	AB
6	Miller	⊥	f	⊥	B
7	Miller	7/7/1959	m	⊥	o
8	Smith	8/9/1970	m	Mass Ave.	⊥
9	Smith	9/8/1970	m	⊥	A

Table 1.10: Minimum Union of the two tables Police and Hospital, handling cases of subsuming tuples.

Complement Union

In this result an influence of complement union is shown

Complement Union of *Police* and *Hospital*

	Name	Birthdate	Sex	Address	Blood
1+7	Miller	7/7/1959	m	234 Main St.	o
2+6	Miller	⊥	f	234 Main St.	B
3+4	Peters	1/19/1953	m	43 First St.	AB
4+5	Peters	1/19/1953	m	⊥	AB
2+7	Miller	⊥	m	234 Main St.	o
8	Smith	8/9/1970	m	Mass Ave.	⊥
9	Smith	9/8/1970	m	⊥	A

Table 1.11: Complement Union of the two example tables, handling cases of complementing tuples.

Summary of fusion operators and strategies discussion

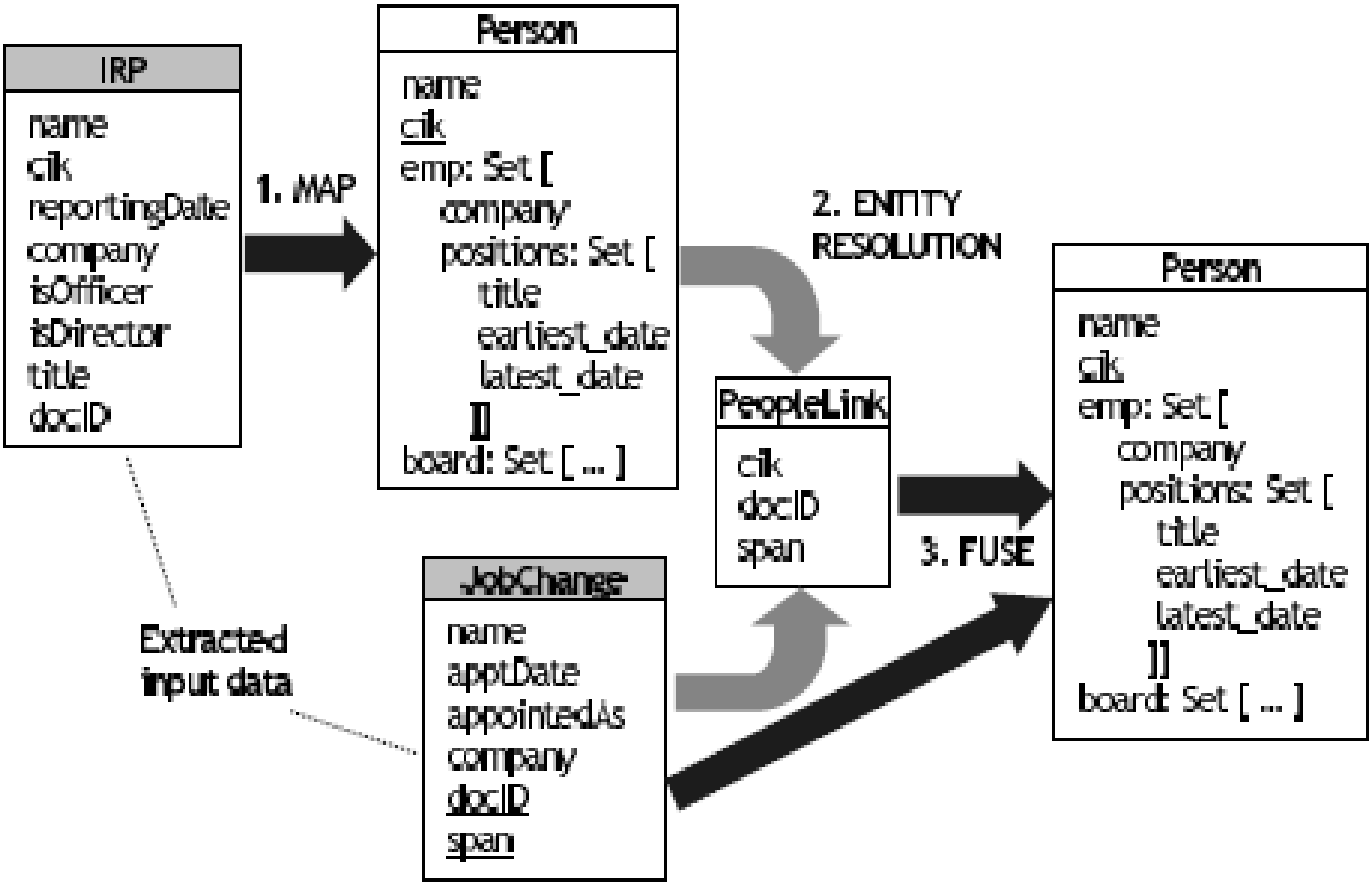
Basic problems remained:

1) Without **strict specification of the expected result of the integration** to make **an analysis of the participating source tables** will be required **to find an adequate data fusion strategy** (many of them are possible: the **situation generally is undecidable** because there is no way to find the best way **of resolving the uncertainties and contradictions** in data)
2) **Better founded matching/deduplication technique is required**
3) **Languages** for specification of data fusion operations (specifically **for virtual data integration**) **are to be investigated** and developed
4) A strategy to **specify the constraints for the result of heterogeneous data fusion** remains not clear; e.g., they should be well agreed with the quality of source data (kind of source constraints ?)
5) **Data fusion strategy for the case when conceptual schema of the result is defined in advance should be investigated**. At least, the problem for data fusion strategy selection looks in such case much clearer (the criteria for the data to be produced are well defined)

Another approach for data fusion

IBM high level language facilities for data fusion

- **The fusion operations considered are not enough**. More flexible facilities are required to be generally applicable for specific cases.
- IBM HIL language provides such possibilities. **Entity resolution rules in HIL are used in conjunction with the entity population rules in order to achieve fusion of the matched or linked entities into the final, integrated entities**.
- The fusion of **complexly structured entities** from large amounts of data is supported (**in contrast to the flat relational model**).
- **Fusion based on links and indexes is provided**.
- **HIL indexes capture the fusion logic in a data integration flow**.



Example Integration Flow

Fusion via HIL rules.

```
rule m5:
      insert  into Employment ! [cik: l.cik]
      select  [ company: j.company,
                positions: Positions ! [cik: l.cik, company: j.company]
              ]
      from JobChange j, PeopleLink l
      where j.docID = l.docID and j.span = l.span and isOfficer (j.appointedAs);

rule m6:
      insert  into Positions ! [cik: l.cik, company: j.company]
      select  [  title: normTitle (j.appointedAs),
                  ...
              ]
      from JobChange j, PeopleLink l
      where j.docID = l.docID and j.span = l.span and isOfficer (j.appointedAs);
```

Union of query terms to map into an entity

```
EmploymentFn :=
   fn (arg). select [ company: i.company,
                     positions: PositionsFn ([ cik: i.cik,
                                               company: i.company]) ]
              from IRP i where arg = [cik: i.cik] and i.isOfficer = true
union
              select [ company: j.company
                     positions: PositionsFn ([ cik: l.cik,
                                               company: j.company]) ]
from JobChange j, PeopleLink l
where j.docid = l.docid and j.span = l.span
 and arg = [cik:l.cik] and isOfficer (j.appointedAs) = true;
```

Example of index manipulation in HIL

```
Person := select [ name: i.name, cik: i.cik,
                   emp: EmploymentFn (cik: i.cik) ]
          from IRP i;
```

To compute the value of emp, we use a function call that corresponds to the index lookup on Employment. The function returns a non-empty set of values only for a finite set of keys.

Another possibility for reference to an index: we will use a **join** with the binary table that materializes the index. The **left outer join** is required. The query always emits an output tuple for each entry in IRP. If there is a match with Employment, as specified by the on clause of the outer join, then e.val is non-null, otherwise we get an emty set (of NULLs) for emp:

```
Person := select [ name: i.name, cik: i.cik, emp: emptyIfNull (e.val) ]
          from IRP i left outer join Employment e
                on [cik: i.cik] = e.key;
```

HIL experience for social large scale data integration

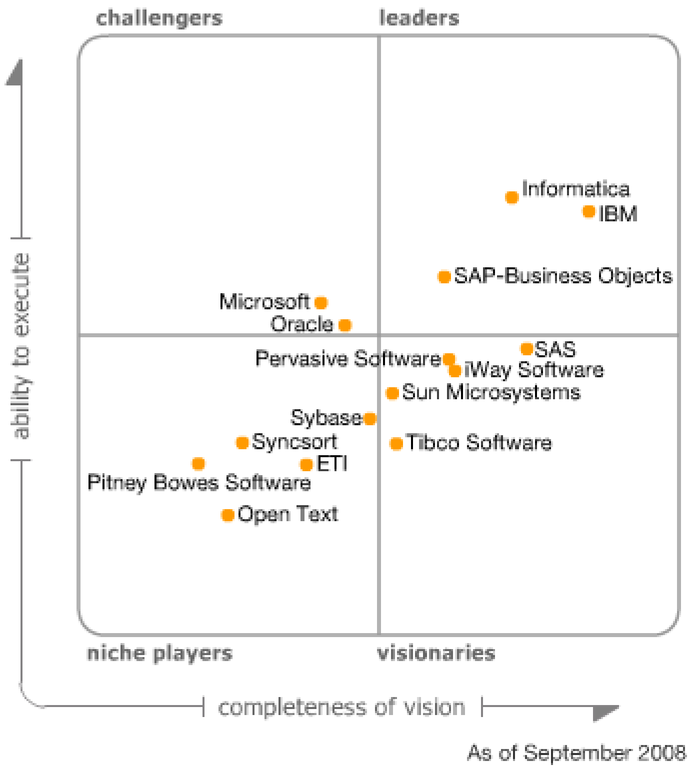
- Twitter for social media: (**25 million of Tweens per day**, **for 6 days 20 millions of entities extracted from texts**, not linearly increasing with time)
- **Each input tweet** (unstructured**) annotated with any number of attributes ranging from personal information** (address, name, marital status, etc.) **to life events** (birthdays, weddings, etc.), **employment** (occupation, employer, etc.), **and shopping-related attributes** (a person's purchase intent or sentiment towards products).
- **A main entity population rule constructs Person profiles,** relying on **12 auxilliary indexes** that accumulate facts about persons across all tweets.
- **Each index is populated by one or more HIL rules**.
- The results were obtained by **running the HIL-generated Jaql queries on a Hadoop cluster with 10 nodes**, each an IBM System x3550, with 2 CPUs (4 cores each), and 32 GB memory
- **To build user profiles** with rich enough historical information, **there were extracted and integrated data that spans months of tweets, with billions of messages.**

HIL language benefits

- HIL (**entity integration** language) bridges **schema mapping and entity resolution**. Associations between entities are explicitly given in HIL via indexes
- HIL indexes **capture the fusion logic in a data integration flow**
- HIL rules provide **explicit resolution actions on constraint violations**, with deterministic results
- HIL is intended **at the incremental modification of target data when new data sources and rules are added**
- **There are several query/transformation languages with complex data processing capabilities** (but not focused on data integration), including XQuery, XSLT, Jaql, and Pig Latin, **which are all possible target languages for HIL compilation.**

Data fusion in existing integration systems

Commercial DI Tools



Source: Gartner

- Typical ETL tools support rule-based fusion
  - IIS (IBM Information Server)
  - SSIS (Microsoft’s SQL Server Integration Services)
  - Etc.

See details in survey [Bleiholder and Naumann, 08

Research DI Systems w. Awareness of Data Conflicts

System	Conflict types	Methodology	Strategy	Specification
Multibase	Schematic, data	Resolution	Choose, Avg, Min, Max, Sum, …	Manually, in query
Hermes	Schematic, data	Resolution	MostRecent, Choose	Manually, in mediator
Fusionplex	Schematic, object, data	Resolution	MostRecent, Min, Max, Avg, …	Manually, in query
HumMer	Schematic, object, data	Resolution	MostAbstract, Vote, Min, ChooseDepen…	Manually, in query
Ajax	Schematic, object, data	Resolution	Various	Manually, in workflow definition
TSIMMIS	Schematic, data	Avoidance	Choose	Manually, rules in mediator
SIMS/Ariadne	Schematic, data	Avoidance	Choose	Automatically
Infomix	Schematic, data	Avoidance	onlyConsistentValue	Automatically
Hippo	Schematic, object, data	Avoidance	onlyConsistentValue	Automatically
ConQuer	Schematic, object, data	Avoidance	onlyConsistentValue	Automatically
Rainbow	Schematic, object, data	Avoidance	onlyConsistentValue	Automatically
Pegasus	Schematic, data	Ignorance	Escalate	Manually
Nimble	Unknown	Ignorance	Escalate	Manually
Carnot	Schematic	Ignorance	Escalate	Automatically
InfoSleuth	Schematic	Ignorance	Escalate	Unknown
Potter's Wheel	Schematic	Ignorance	Escalate	Manually, transformation

See details in survey [Bleiholder and Naumann, 08]

Other DI Systems

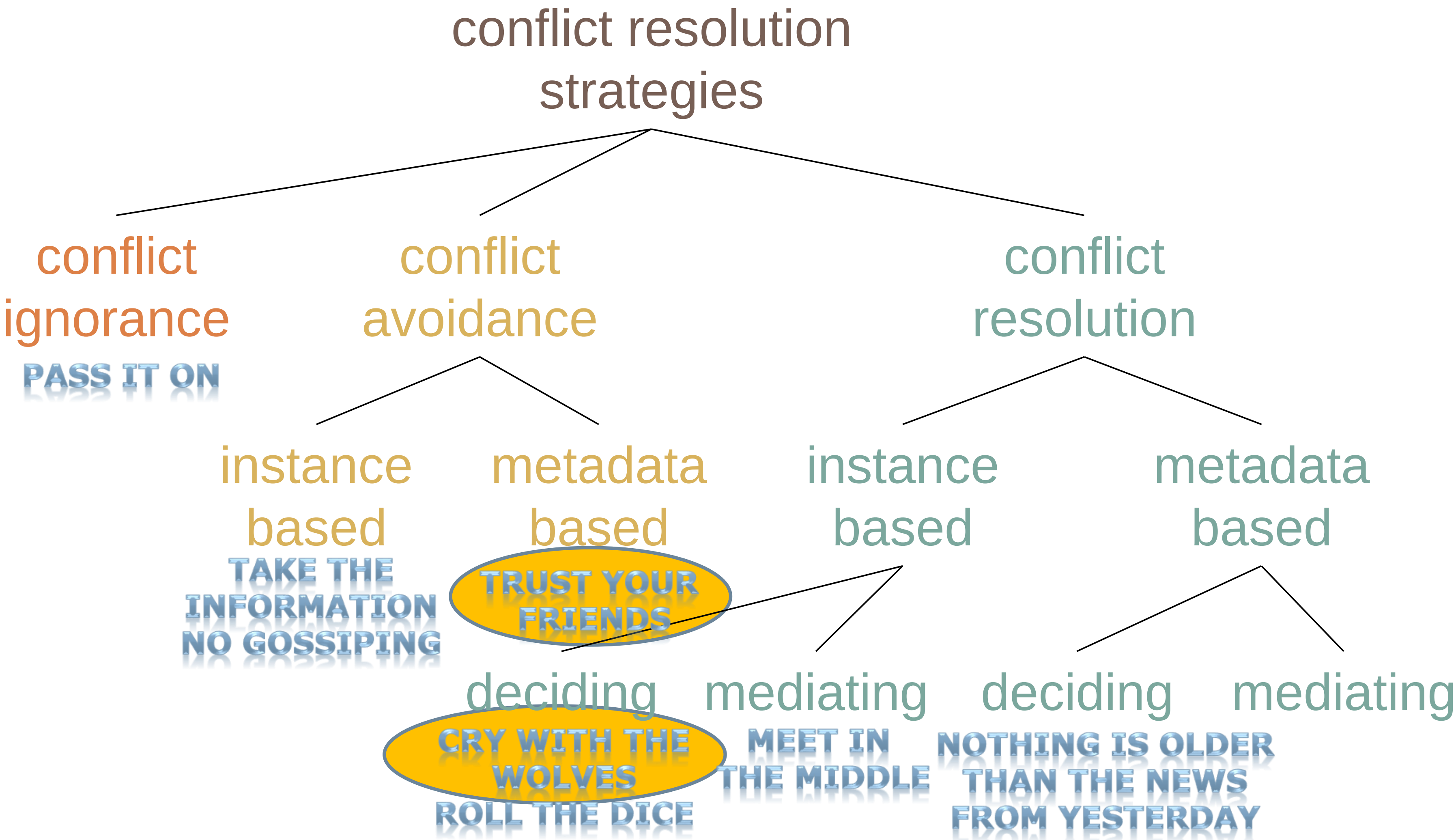
- Research DI systems
  - Trio: including accuracy and lineage into data model
  - Information Manifold
  - Garlic
  - Disco (Distributed Information Search Component)
  - Synthesis
  - etc.
- Peer data management systems
  - Orchestra: allowing multiple viewpoints
  - Hyper: isolating the minimum amount of data to reach consistency

Truth-discovery techniques

What else is there?

- Consistent Query Answering
  - Avoid conflicts and report only certain tuples
    - Those that appear in every repair [FFM05]
- “Possible worlds” models
  - Build all possible solutions, annotated with likelihood
    - Yes/No/Maybe [DeM89]
    - Probability value [LSS94]
  - Probabilistic databases [SD05]
    - Extend algebra to produce probabilities
    - Extend query language to query and export probabilities

Basic Strategies



Intuitions

- Data sources are of different quality and we trust data from accurate sources more
- The real world is dynamic and the true value often evolves over time
  - E.g., person affiliation, business contact phone
- Data sources can copy from each other and errors can be propagated quickly

Other Considerations

- **Accuracy**
  - Some data sources are more trustworthy than others
  - Consider accuracy when deciding true values
- **Freshness**
  - World changes dynamically
  - A value can be true or false, but can also be out-of-date
  - Distinguish between incorrect and out-of-date
- **Dependency**
  - Data sources can copy (parts) from other data sources
  - Consider this dependency

Advanced Truth-Discovery Techniques

- Data sources are of different quality and we trust data from accurate sources more → Consider accuracy of sources
- The real world is dynamic and the true value often evolves over time → Consider freshness of sources
  - E.g., person affiliation, business contact phone
- Data sources can copy from each other and errors can be propagated quickly → Consider dependence between sources

Advanced Truth-Discovery Techniques

- Data sources are of different quality and we trust data from accurate sources more
- The real world is dynamic and the true value often evolves over time
  - E.g., person affiliation, business contact phone
- Data sources can copy from each other and errors can be propagated quickly

Consider accuracy of sources

Consider freshness of sources

Consider dependence between sources

Trust Accurate Sources

- Considering accuracy can often improve truth discovery

	S1	S2	S3
Stonebraker	MIT	Berkeley	MIT
Dewitt	MSR	MSR	UWisc
Bernstein	MSR	MSR	MSR
Carey	UCI	AT&T	BEA
Halevy	Google	Google	UW

Trust Accurate Sources

- Considering accuracy can often improve truth discovery

	S1	S2	S3
Stonebraker	MIT	Berkeley	MIT
Dewitt	MSR	MSR	UWisc
Bernstein	MSR	MSR	MSR
Carey	UCI	AT&T	BEA
Halevy	Google	Google	UW

S1 is more accurate; trusting it more can help find the correct affiliation for Carey

Trust Accurate Sources

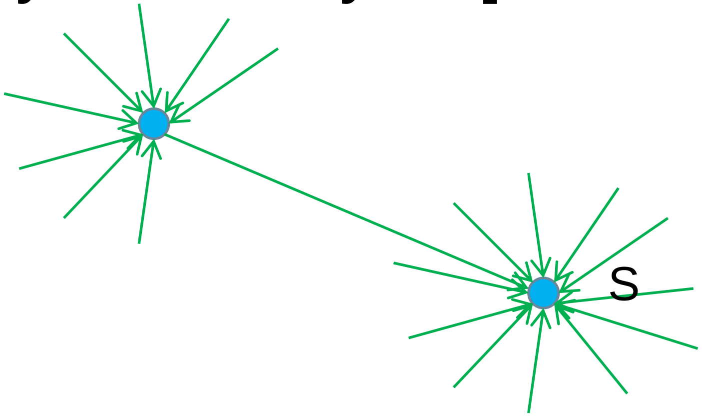
- Considering accuracy can often improve truth discovery

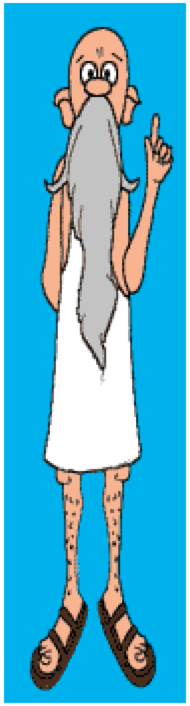
	S1	S2	S3
Stonebraker	MIT	Berkeley	MIT
Dewitt	MSR	MSR	UWisc
Bernstein	MSR	MSR	MSR
Carey	UCI	AT&T	BEA
Halevy	Google	Google	UW

S1 is more accurate; trusting it more can help find the correct affiliation for Carey

Find Trustable Sources (I)

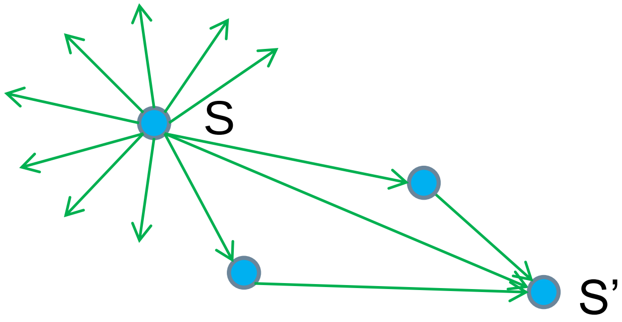
- Deciding authority based on link analysis and source popularity
  - Survey: “Link analysis ranking: algorithms, theory, and experiments” [Borodin et al., 05]
  - PageRank [Brin and Page, 98]
  - Authority-hub analysis [Kleinberg, 98]





Find Trustable Sources (II)

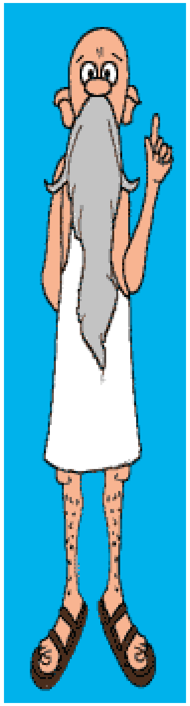
- Assign a global trust rating to each data source based on its behavior in a P2P network
  - TrustMe [Singh and Liu, 03]
  - EigenTrust [Kamvar et al., 03]
    - Peer i&j:



$$s_{ij} = sat(i, j) - unsat(i, j)$$

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(S_{ij}, 0)}$$

$$t_{ij} = \sum_k c_{ik} c_{kj}$$

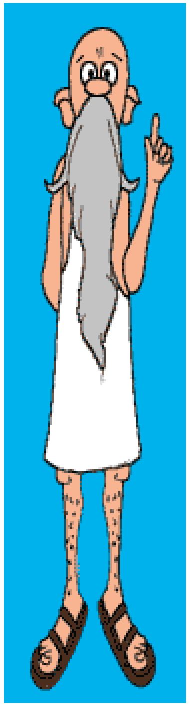


Find Trustable Sources (III)

- Compute accuracy of sources
  - Corroborating answers from web sources [Wu and Marian, 07]
  - TruthFinder [Yin et al., 07]
  - Solomon [Dong et al., 09a]

$$A(S) = \underset{v \in \overline{V}(S)}{Avg}\, P(v)$$

$\overline{V}(S)$-values provided by S; P(v)-pr of value v being true



Model and Algorithm

Properties

- A value provided by more accurate sources has a higher probability to be true
- Assuming uniform accuracy, a value provided by more sources has a higher probability to be true

Source accuracy

$$A(S) = \underset{v \in \bar{V}(S)}{Avg}\, P(v)$$

Source trustworthy

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

Value confidence

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

Value probability

$$P(v) = \frac{e^{C(v)}}{\sum_{v_0 \in D(O)} e^{C(v_0)}}$$

- Consider value similarity

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

- Continue until source accuracy converges

Advanced Truth-Discovery Techniques

- ☑ Data sources are of different quality and we trust data from accurate sources more
- ☐ The real world is dynamic and the true value often evolves over time
  - E.g., person affiliation, business contact phone
- ☐ Data sources can copy from each other and errors can be propagated quickly

Consider accuracy of sources

Consider freshness of sources

Consider dependence between sources

A Dynamic World

- True values can evolve over time
  - A subtle third case: *out-of-date*

	S1	S2	S3
Stonebraker	(03, MIT)	(00, Berkeley)	(06, MIT)
Dewitt	(09, MSR)	(08, MSR)	(01, UWisc)
Bernstein	(00, MSR)	(00, MSR)	(01, MSR)
Carey	(09, UCI)	(05, AT&T)	(06, BEA)
Halevy	(07, Google)	(05, Google)	(06, UW)

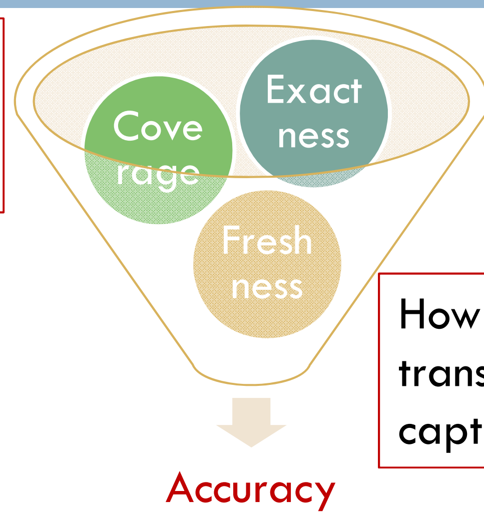
A Dynamic World

- True values can evolve over time
  - A subtle third case: *out-of-date*
- Low-quality data can be caused by different reasons

	S1	S2	S3
Stonebraker (Θ, Berkeley), (02, ***MIT***)	**(03, MIT)**	**(00, Berkeley)** OUT-OF-DATE!	(01, Berkeley) **(06, MIT)**
Dewitt (Θ, UWisc), (08, ***MSR***)	(00, UWisc) **(09, MSR)**	(00, UW) (01, UWisc) **(08, MSR)**	**(01, UWisc)** OUT-OF-DATE!
Bernstein (Θ, ***MSR***)	**(00, MSR)**	**(00, MSR)**	**(01, MSR)**
Carey (Θ, Propell), (02, BEA), (08, ***UCI***)	(04, BEA) **(09, UCI)**	**(05, AT&T)** ERR!	**(06, BEA)** OUT-OF-DATE!
Halevy (Θ, UW), (05, ***Google***)	(00, UW) **(07, Google)**	(00, UWisc) (02, UW) **(05, Google)**	(01, UWisc) **(06, UW)** SLOW!

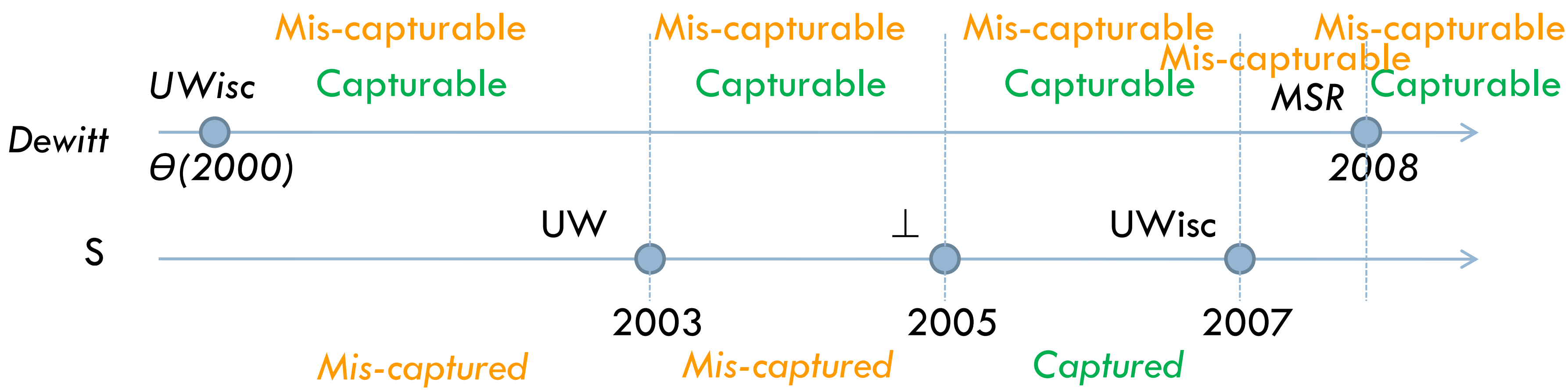
Refine Accuracy of Sources [Dong et al., 09b]

How many transitions are *captured*



How many transitions are not *mis-captured*

How quickly transitions are captured



Coverage = #*Captured*/#*Capturable* (e.g., ¼=.25)

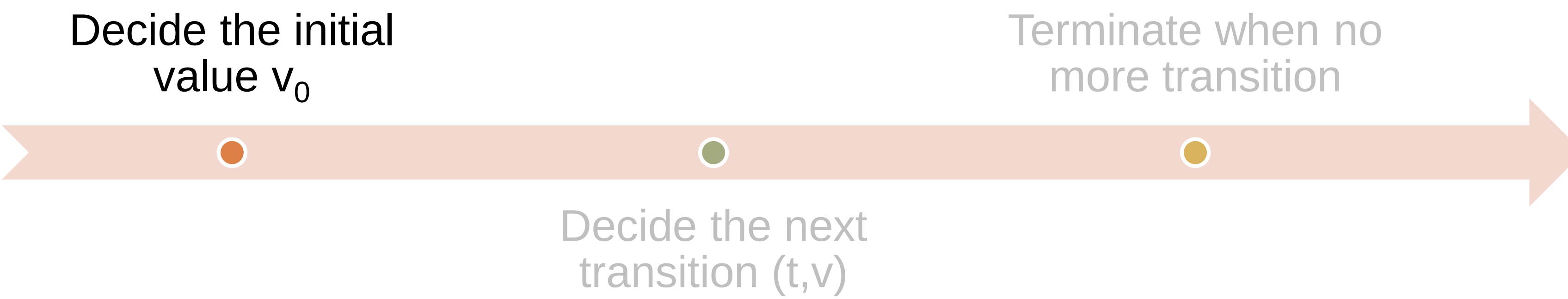
Exactness = 1-#*Mis-Captured*/#*Mis-Capturable* (e.g., 1-2/5=.6)

Freshness(Δ) = #(*Captured w. length*<=Δ)/#*Captured* (e.g., F(0)=0, F(1)=0, F(2)=1/1=1…)

Freshness Measures in Other Work

- Other work on data freshness: Compare a materialized view with the original source
  - [Peralta, Ph.D. Thesis'06]: timeliness, currency
  - [Guo et al., 05]: completeness, consistency, currency
  - [Olston and Widom, 05]: divergence
  - [Labrinidis and Roussopoulos, 04]: QoD(freshness)
  - [Theodoratos and Bouzeghoub, 01]: consistency
  - [Cho and Garcia-Molina, 00]: freshness, age

Discover Evolving True Values



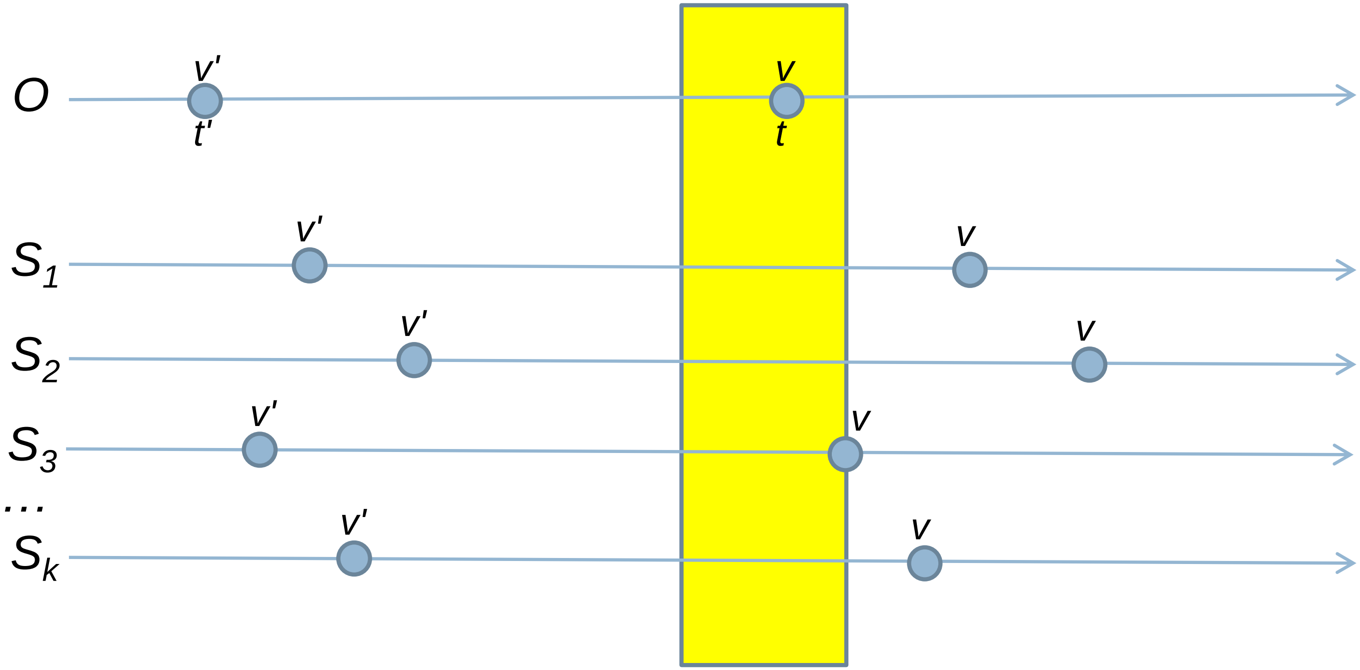
- Decide the initial value: according to the Bayes Rule, we need to know
  - $Pr(\Phi(S)|v_i)$ for each value v_i
    - If S provides v_i : E(S)C(S)
    - If S does not provide any value: E(S)(1-C(S))
    - If S provides another value: (1-E(S))/n
  - $Pr(\Phi(S)|\perp)$—the object does not exist initially
    - If S does not provide any value: E(S)
    - If S provides a value : (1-E(S))/(n+1)

Discover Evolving True Values

Decide the initial value v_0

Terminate when no more transition

Decide the next transition (t,v)



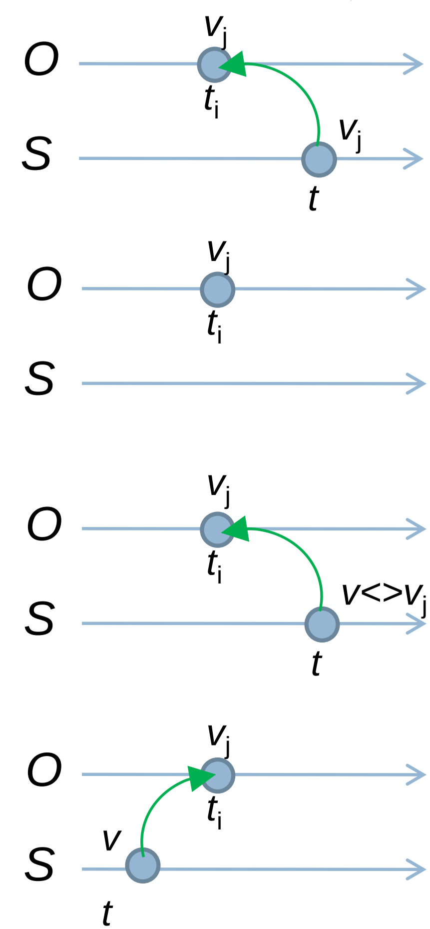
Discover Evolving True Values

Decide the initial value v_0

Decide the next transition (t,v)

Terminate when no more transition

- Decide the next transition (t,v): according to the Bayes Rule, we need to know
  - $Pr(\Phi(S)|(t_i,v_j))$ for each time t_i and value v_j
    - If S provides v_j at time t : $E(S)C(S)F(S, t-t_i)$
    - If S does not update any more: $E(S)(1-C(S)F(S, t_n-t_i))$
    - If S makes a wrong update: $(1-E(S))/n(t_n-t')$ (t_n—the last obs point, t'—time of the prev update)
  - $Pr(\Phi(S)|\text{no more transition})$: similarly computed
    - If S does not update any more: $E(S)$
    - If S makes an update: $(1-E(S))/(n+1)(t_n-t')$



Advanced Truth-Discovery Techniques

- ☑ Data sources are of different quality and we trust data from accurate sources more
- ☑ The real world is dynamic and the true value often evolves over time
  - E.g., person affiliation, business contact phone
- ☐ Data sources can copy from each other and errors can be propagated quickly

Consider accuracy of sources

Consider freshness of sources

Consider dependence between sources

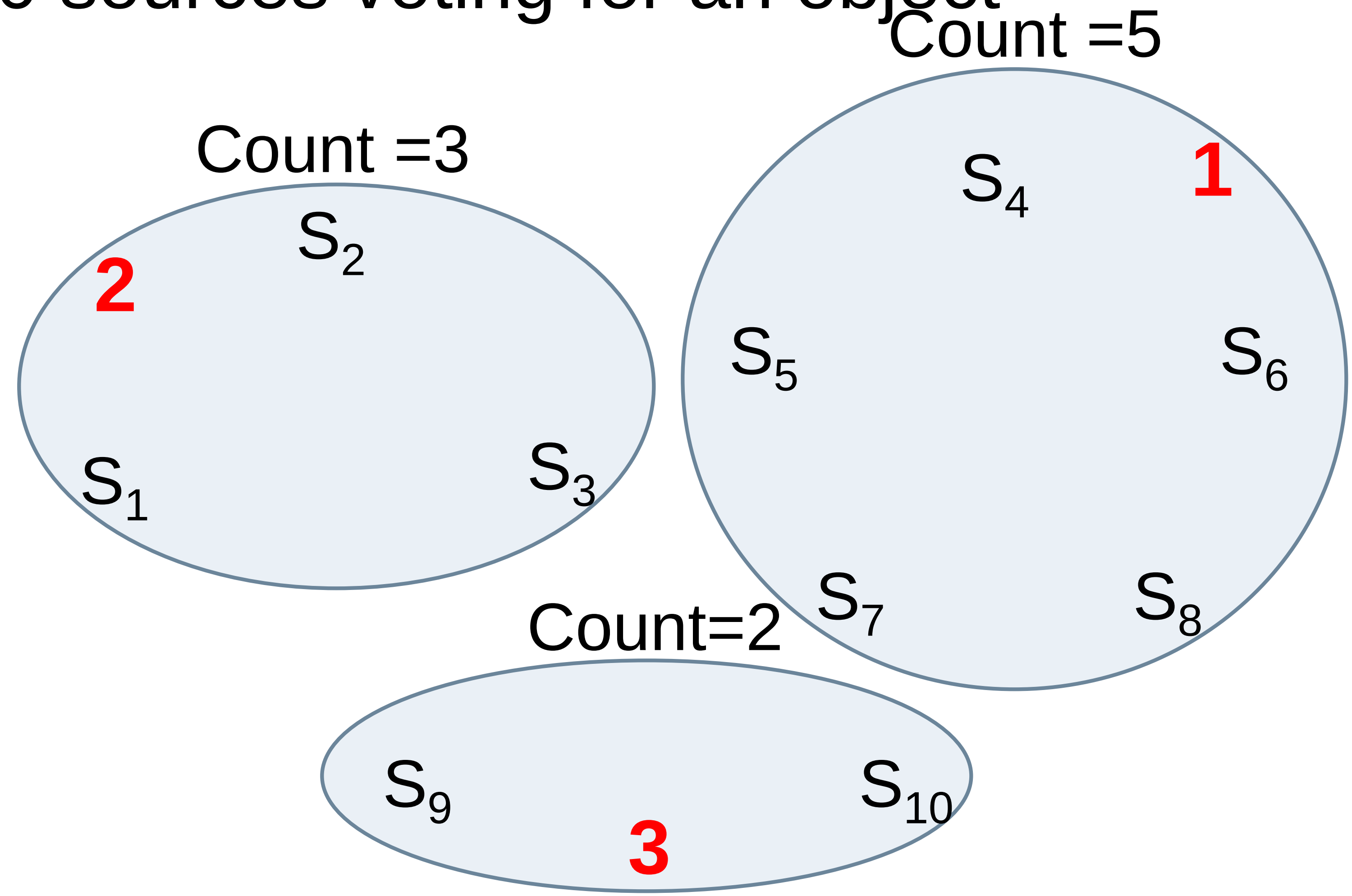
Copied Data Can Change Truth Discovery Results

- Previous methods assume source independence

	S1	S2	S3	S4	S5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

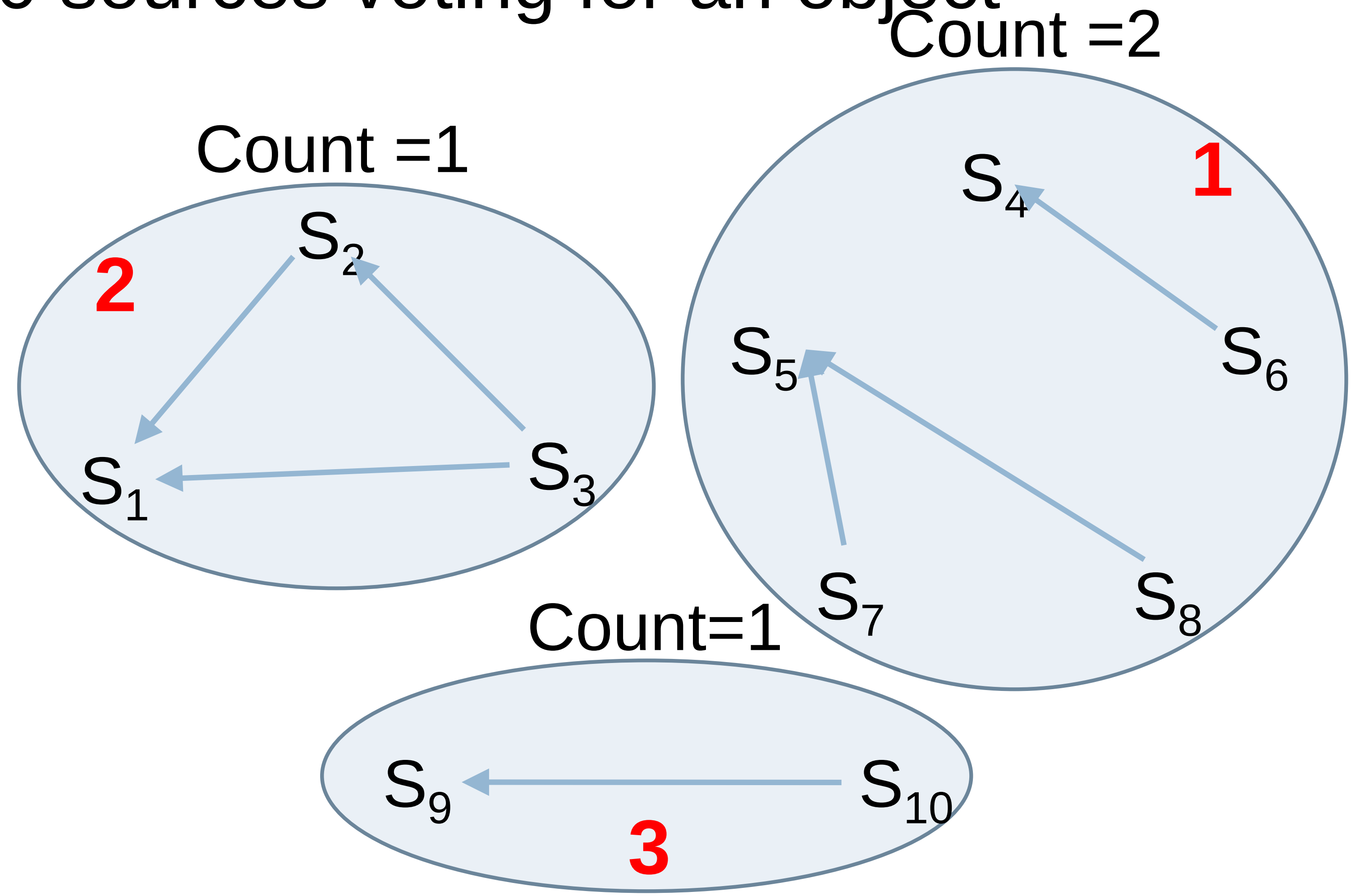
Error propagation by copying
Voting for Independence Sources

- 10 sources voting for an object



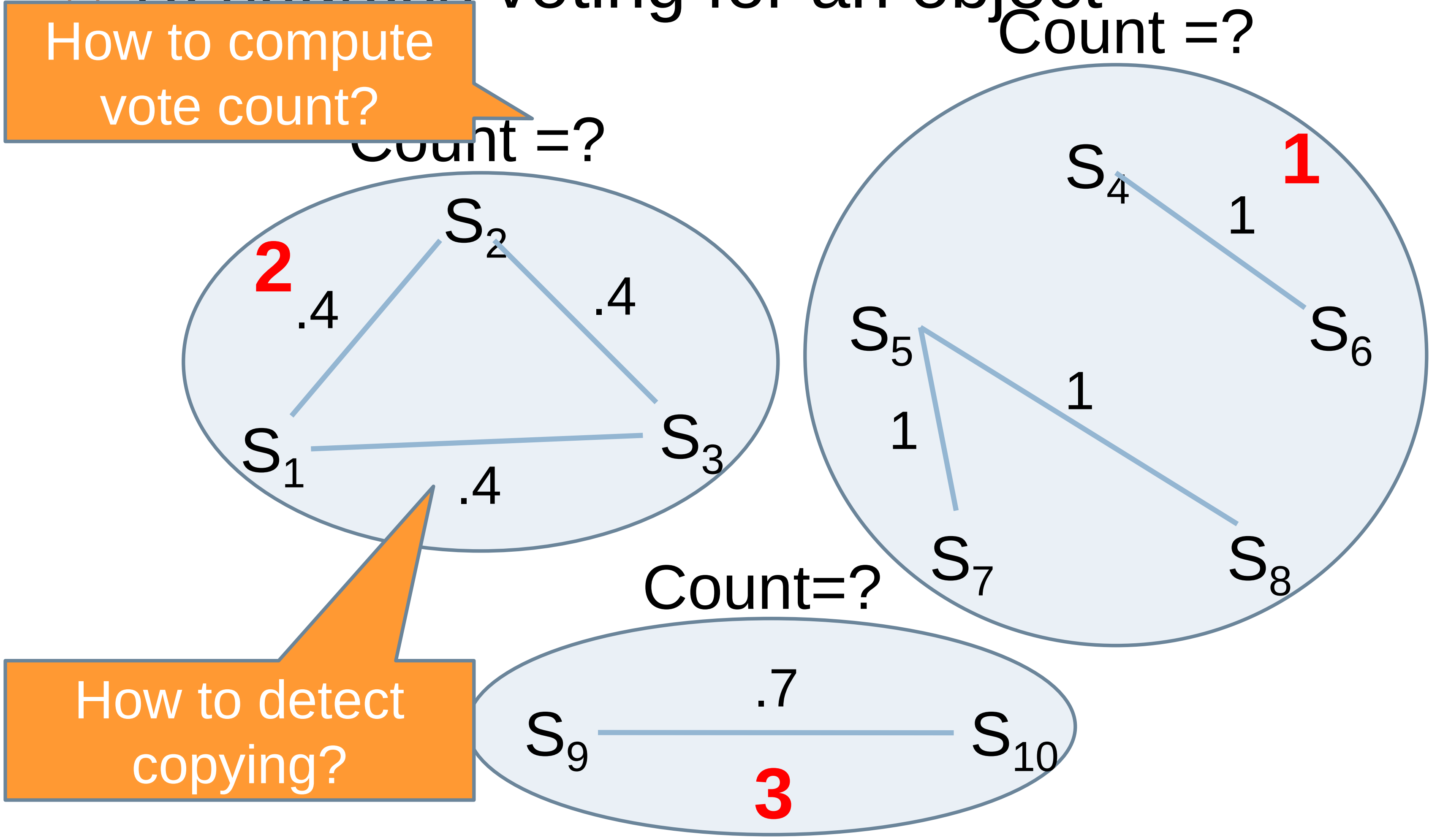
Voting w. Knowledge of Copying

- 10 sources voting for an object



Voting w. Probabilistic Copying

- 10 sources voting for an object



Considering Dependence

- Opinion pooling: combine probability distribution from multiple experts
  - Combination of opinions [Chang, Ph.D. thesis'85]
  - Reconciliation of probability distributions [Lindley, 83]
  - Updating of belief in the light of someone else's opinion [French, 80]
- Data fusion w. source dependence
  - [Dong et al., 09a][Dong et al., 09b]

Open Problems and Challenges

Open Problems

- Accuracy of fusion
- Efficiency of fusion
- Usability of fusion
- Interaction with other components of data integration

Accuracy of Fusion (I)

- **Challenge 1:** Correlated values
  - E.g.1, (firstName, lastName) from 4 sources
    - S1: (Xin, Dong)
    - S2: (Xin Luna, Dong)
    - S3: (Dong, Xin)
    - S4: (Dong, Xin Luna)

  Voting → (Dong, Dong)

  - E.g.2, (ISBN, authors) from 3 sources
    - S1: (**1, Peter Loshin) (**2, Peter Loshin)
    - S2: (**1, Pete Loshin)
    - S3: (**1, Pete Loshin)

  Voting → (**1, Pete Loshin)
  (**2, Peter Loshin)

- **Current effort:** ChooseDepending(val, col)
- **Directions:** consider correlation at the attribute level and at the instance level.

Accuracy of Fusion (II)

- **Challenge 2:** Different formating styles
  - E.g., (ISBN, authors) from 4 sources

Src1
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang, Roy A Plastock)
(**4, Peter Aiken, David M Allen)…

Src2
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang, Roy Plastock)
(**4, Peter Aiken, David Allen)…

Skip middle-names

Src3
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang)
(**4, David Allen, Peter Aiken)…

Accuracy of Fusion (II)

- **Challenge 2:** Different formating styles
  - E.g., (ISBN, authors) from 4 sources

Src1
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang, Roy A Plastock)
(**4, Peter Aiken, David M Allen)…

Src2
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang, Roy Plastock)
(**4, Peter Aiken, David Allen)…

Skip middle-names

Src3
(**1, Pete Loshin)
(**2, Dennis Suhanovs)
(**3, Zhigang Xiang)
(**4, David Allen, Peter Aiken)…

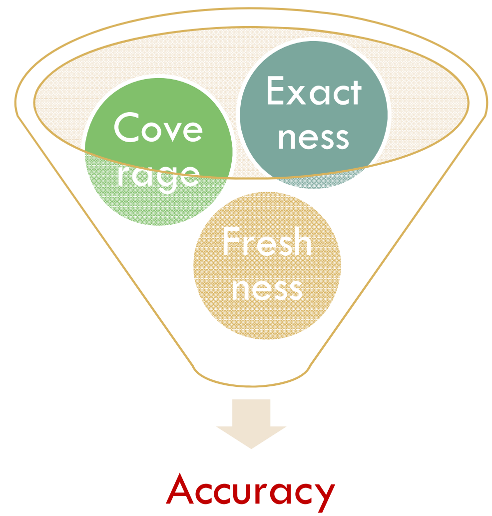
Src4
(**1, Pete Loshin)
(**2, Dennis uhanovs)
(**3, Zhigang Xiang)
(**4, Peter Aiken)…

Only first-authors

- **Current effort:** consider value similarity
- **Directions:** consider formatting styles used by each source.

Accuracy of Fusion (III)

- **Challenge 3:** Source profiling
  - **Current effort:** accuracy (coverage, exactness, freshness)
  - Data properties can be different for different categories of data
    - Source A is a vertical source on restaurants
    - Source B knows very well about NYC
  - Data properties can evolve over time
    - Source C improves its data over time
- **Directions:** partition data into different portions and profile on each portion



Efficiency of Fusion (I)

- **Challenge 4:** Incremental fusion
  - When we have more data sources (e.g., Src4) or lose some data sources, shall we do data fusion from scratch?

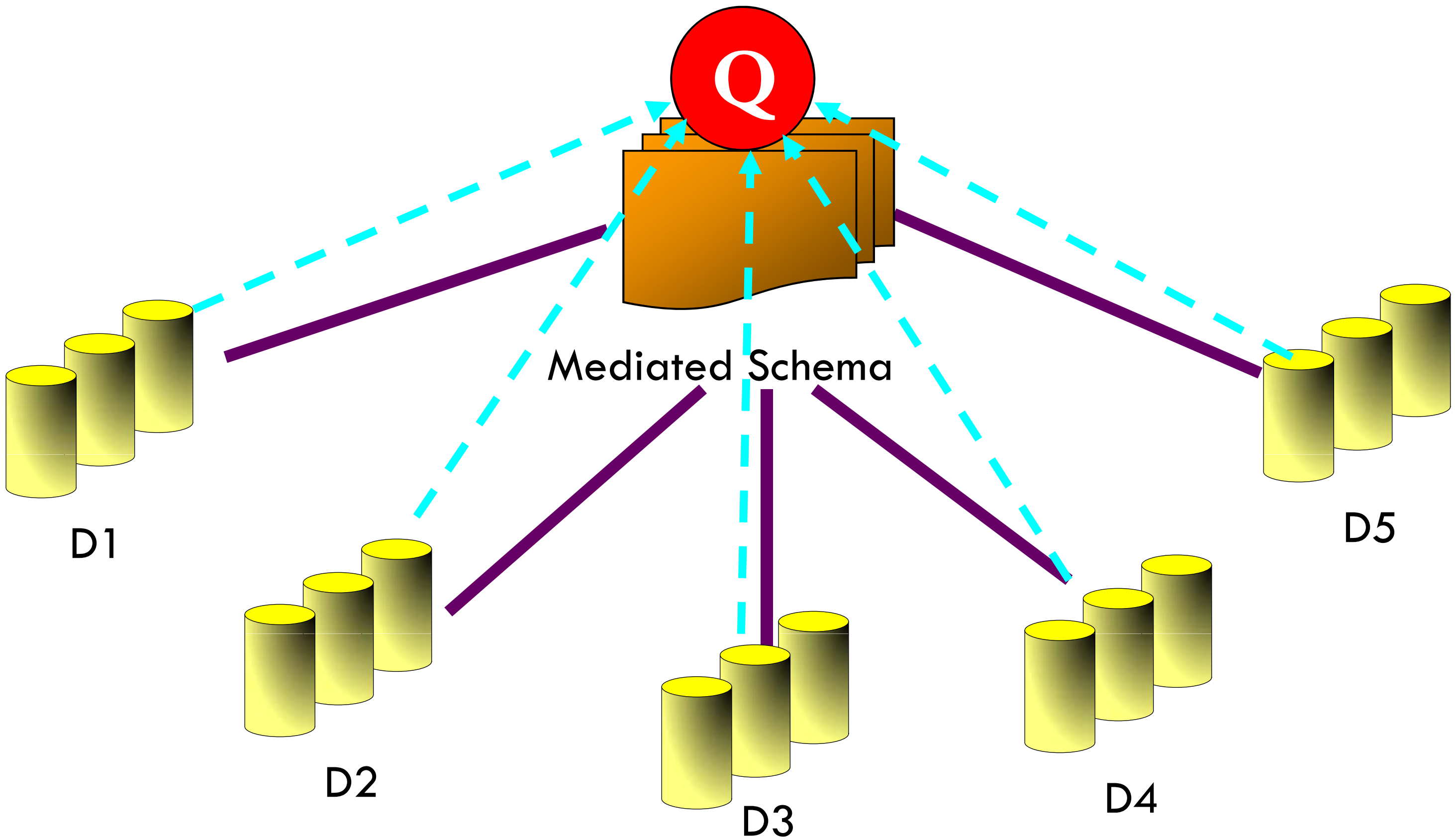
Src1	Src2
(**1, Pete Loshin)	(**1, Pete Loshin)
(**2, Dennis Suhanovs)	(**2, Dennis Suhanovs)
(**3, Zhigang Xiang, Roy A Plastock)	(**3, Zhigang Xiang, Roy Plastock)
(**4, Peter Aiken, David M Allen)…	(**4, Peter Aiken, David Allen)…

Src3	Src4
(**1, Pete Loshin)	(**1, Pete Loshin)
(**2, Dennis Suhanovs)	(**2, Dennis uhanovs)
(**3, Zhigang Xiang)	(**3, Zhigang Xiang)
(**4, David Allen, Peter Aiken)…	(**4, Peter Aiken)…

  - When more data come, shall we start from scratch?
- **Directions:** maintain metadata or statistics, retain data lineage.

Efficiency of Fusion (II)

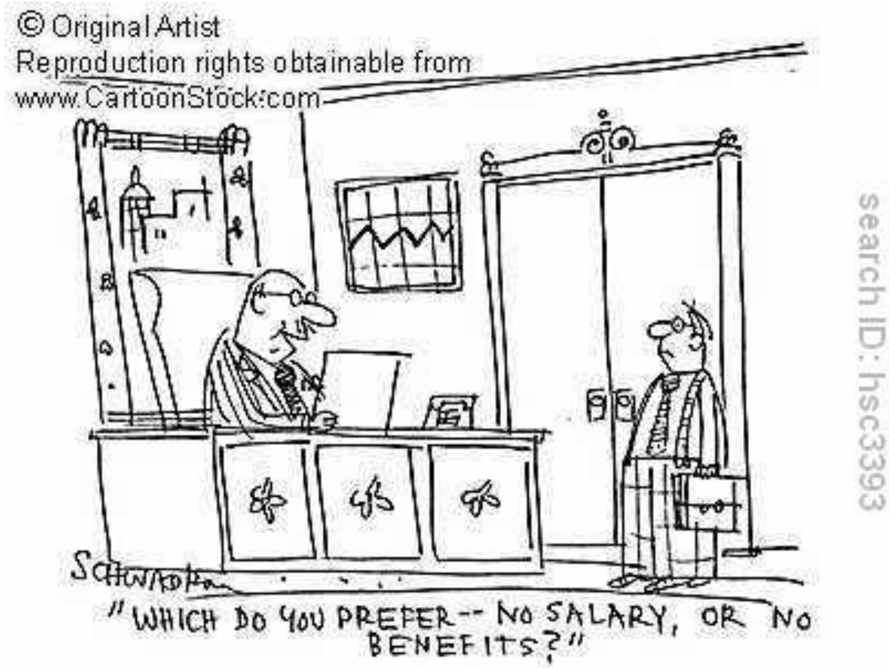
- **Challenge 5:** Runtime fusion
  - In some applications fusing data upfront is infeasible



- **Directions:** maintain source profiles by sampling; emphasize efficiency.

Usability of Fusion (I)

- **Challenge 6**: Personalized fusion
  - Express preference on certain sources
  - Emphasize certain property; e.g., up-to-date vs. high coverage
  - Use certain formats; e.g., full author list vs. only first author
- **Current effort:**
  - Function choose(src)
  - Operator Prioritized-Merge
- **Directions**:
  - A language to express such user preferences
  - Algorithms for efficient execution.

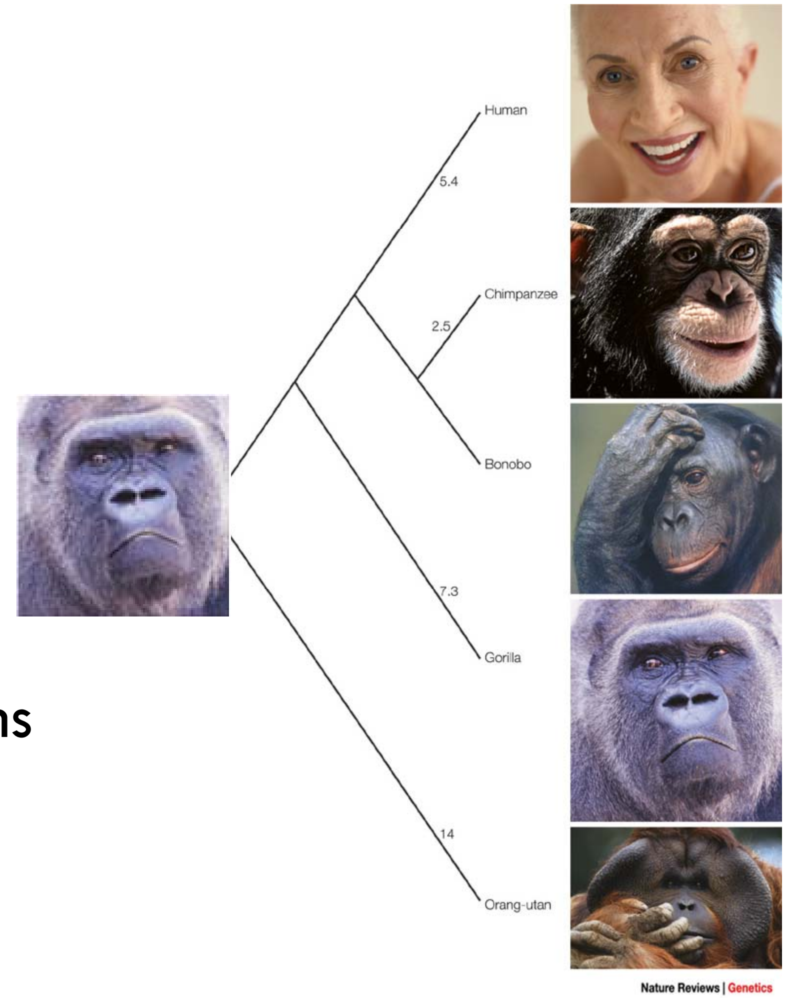


Usability of Fusion (II)

- **Challenge 7**: User feedback
  - Correct certain errors
- **Directions**:
  - Critical questions that can best improve the fusion results
  - A way for users to browse source data and fusion results, and correct mistakes
  - Quickly fixing errors and propagation to related items

Usability of Fusion (III)

- **Challenge 8:** Data Lineage
  - Legal requirement
  - Application requirement: e.g., fusing two customers
  - HCI requirement: HOW did you merge the data? And WHY?
- **Directions:**
  - Effective representation of lineage information
  - Explanation of merging decisions
  - Effective way to find disappeared data items
  - Reversibility and repeatability



Interaction with Other Components of DI (I)

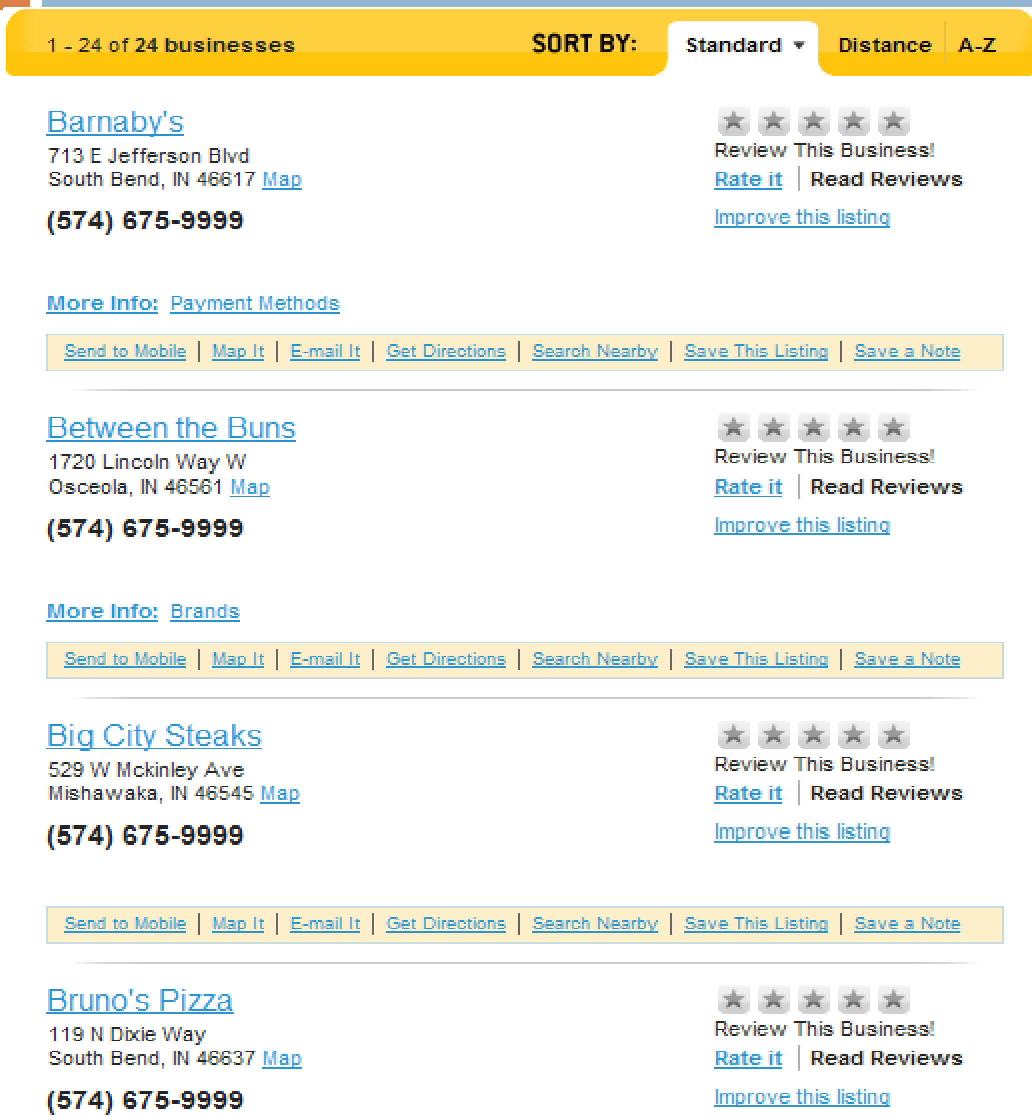
- **Challenge 9**: Fuse data w. different schemas
  - E.g., Contact information from three sources
    - S1: (pid = “1”, work phone = “1234”, home phone = “8765”, mobile phone = “4321”)
    - S2: (pid = “1”, daytime phone = “1234”, evening phone = “4321”)
    - S3: (pid = “1”, phone = “4321”)
- **Directions:** Combine data fusion w. schema matching

Interaction with Other Aspects of DI (II)

- **Challenge 10**: Distinguish wrong values from alternative representations of correct values

- **Directions:** Combine data fusion w. record linkage

A Quiz



Which type of listing are they?

A: are the same business

B: are different businesses sharing the same phone#

C: are different businesses, only one with correct phone#

Conclusions

- Foundations
  - Strategies and functions
  - Operators
- Advanced techniques
  - Consider accuracy
  - Consider freshness
  - Consider dependence
- Open problems
  - Accuracy
  - Efficiency
  - Usability
  - Interaction with other components of D