

Autoplex: Automated Discovery of Content for Virtual Databases

Jacob Berlin and Amihai Motro

Information and Software Engineering Department

George Mason University, Fairfax, VA 22030

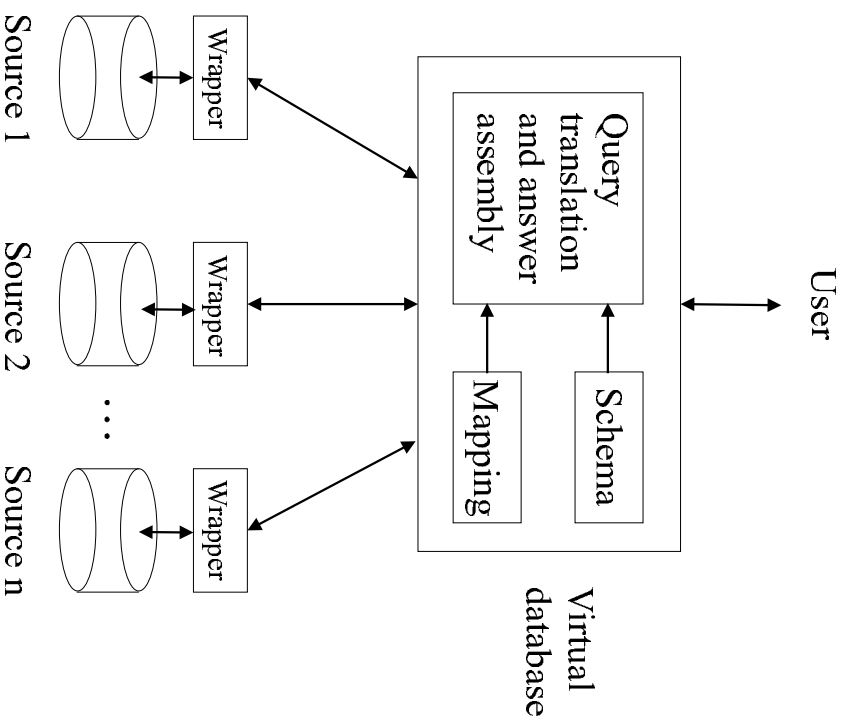
{jberlin, ami}@gmu.edu

September 6, 2001

Database Integration: Background

- Objective is to provide transparent, flexible, and efficient access to heterogeneous databases.
- “Ancient” research issue (20⁺ years of extensive work).
- Focus has been on scheme integration and more recently, data inconsistency.
- Applications include heterogeneous databases, data warehousing, electronic commerce, semantic query processing, and much more.

Database Integration: Typical Architecture



An Example System: Multiplex

- A global (virtual) database scheme is constructed in the relational model *independent* of source schemes.
- Mappings are not inherent; they must be volunteered.
- Each mapping consists of a pair of views: one the local database and one on the global database.
- Mappings can be added and removed continuously and dynamically.

Current Limitations

- Requires experts to manually produce the mappings.
- Effort is linear in the number of sources. Integrating the next source is not any easier than the previous source.
- Works under the assumption that the community of member databases is small and stable.
- Mapping process does not scale up to an environment with a very large number of databases.
- **Bottom line:** Prohibitive costs of mapping new sources.

Solution: Machine Learning

- The integration of heterogeneous databases with a global scheme can be cast as a supervised machine learning problem.
- Specifically, the task of discovering content in new information sources can be learned from examples that are already mapped into the global scheme.

Two Main Problems

1. **Locate** a candidate database. Maybe a set of legacy databases or sources off the world wide web.
2. **Search** the candidate for an acceptable contribution (a transformation of the candidate that fits into the global database).

We focus on 2.

Formalization of the Problem

Given the following virtual database environment:

1. A relation scheme $R = (X_1, \dots, X_n)$. This is the virtual database. Each column X_i of R is labeled as either *required* or *optional*.
2. A set of contribution examples, each consisting of a relation scheme $S = (Y_1, \dots, Y_k)$, a relation instance s of scheme S , and a relational algebra expression e on relation S that defines a contribution to R .
3. A new, previously unseen relation scheme $T = (Z_1, \dots, Z_m)$ and a relation instance t of T . We shall often refer to T as a *candidate* relation.

Formalization of the Problem (continued)

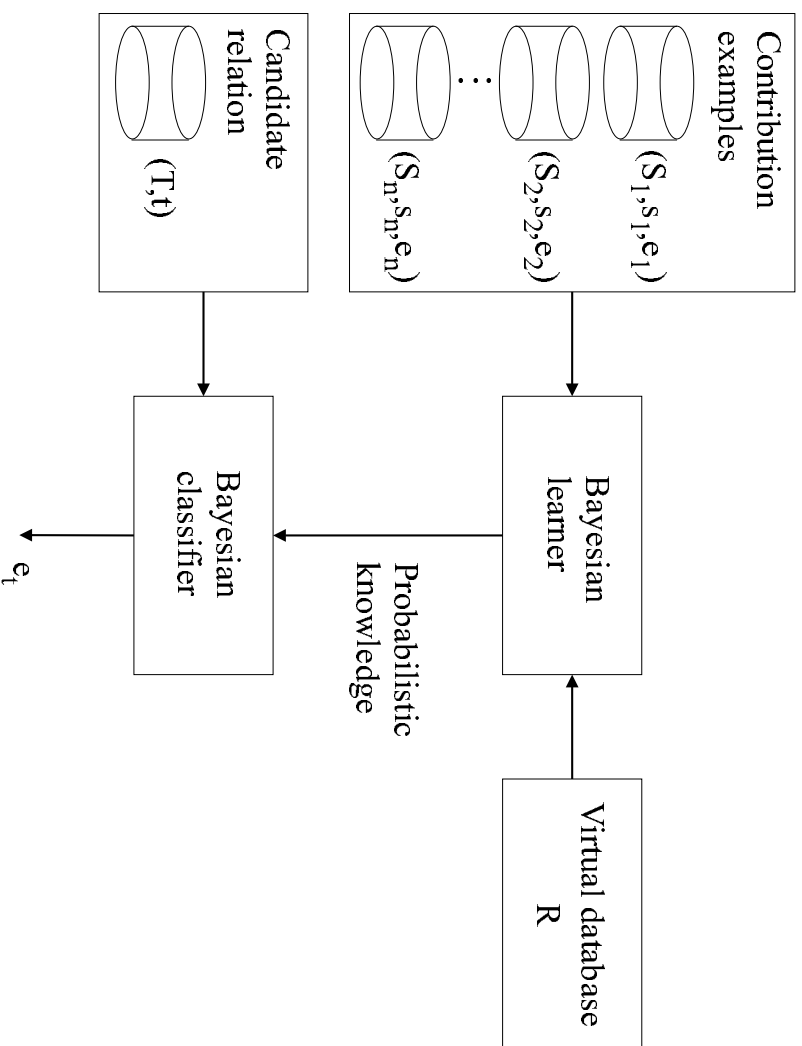
Determine:

- Whether T contains an acceptable contribution to R , and if so, find the relational algebra expression e_t that defines it. An acceptable contribution is one that satisfies all required columns in R and exceeds a predetermined threshold.

Current Limitation:

- The example expressions and the discovered expression are selection-projection operations.

Autoplex Architecture



Bayesian Learning and Classification

- The best classification is the one that is most probable.
- Bayes theorem:
$$P(c|D) = \frac{P(D|c)P(c)}{P(D)}$$

- Assume conditional independence of data values given the classification, thus
$$P(D|c) = P(d_1|c)P(d_2|c) \dots P(d_n|c).$$

- **Learning:** Estimate needed probabilities by counting feature occurrences in examples.

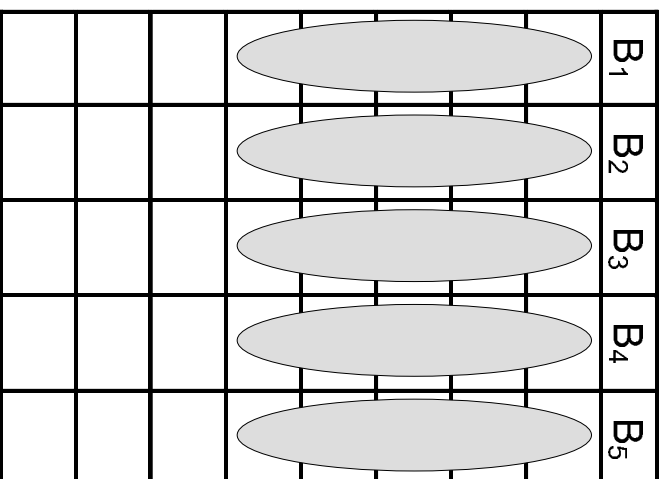
- **Classification:** Find the $c_i \in C$ such that $\forall c_j \in C$ and $c_j \neq c_i$,

$$P(c_i) \prod_{k=1}^n P(d_k|c_i) \geq P(c_j) \prod_{k=1}^n P(d_k|c_j)$$

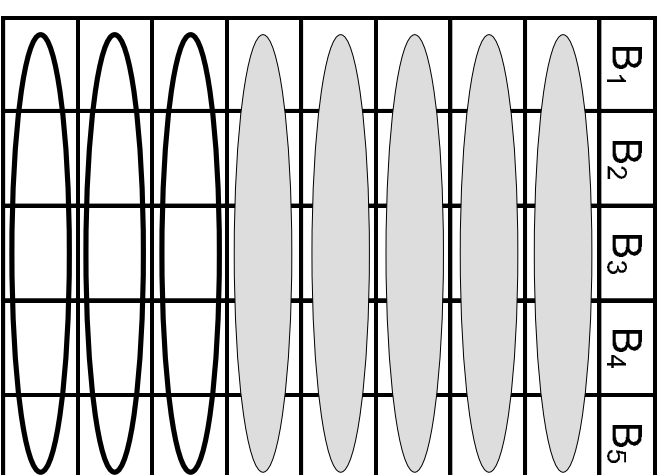
Learning from Examples

Learning from the examples is accomplished in two phases:

Phase 1



Phase 2



Shaded Ovals: Data in projection and selection of e

Clear Ovals: Data in projection but discarded by selection of e

Learning Column Features

- Analogous to *document classification*.
- A local database column is a “document” and global column name is the document classification.
- Inputs to the column learner are:
 1. Documents (column values).
 2. Document classifications (column names).
- Classifier must answer “Does document C_i match class A_j ?”
 1. C_i is an arbitrary column in a candidate table.
 2. A_j is an arbitrary column in the virtual table.
- Employ one learner and classifier for each document class.

Learning Row Features

- Analogous to the “Play Tennis” machine learning problem: Given a tuple describing the weather conditions, would we play tennis?
- Transpose to: Given a tuple from a candidate table, should the tuple be accepted into the virtual table?
- Inputs to the row learner are:
 1. Tuples of values from examples.
 2. Label representing whether this tuple contributes to the virtual table.

A Simple Learning Example

Consider this example relation that is mapped to our global database:

Outlook	Temp	Humidity	Wind
sunny	87	low	5
sunny	90	low	3
overcast	86	normal	2
overcast	80	normal	10
rainy	70	high	11
rainy	75	high	9

} Selected Tuples

} Discarded Tuples

Classification: Finding Content in a Candidate

Step 1 of 4

Calculate probabilities of all possible column matches. Output is a bipartite graph.

Let $R = (A_1, A_2, A_3, A_4)$ and $T = (C_1, C_2, C_3, C_4, C_5, C_6)$

	A_1	A_2	A_3	A_4
C_1	0.3 (0.7)	0.3 (0.7)	0.6 (0.4)	0.0 (1.0)
C_2	0.8 (0.2)	0.9 (0.1)	0.7 (0.3)	0.0 (1.0)
C_3	0.6 (0.4)	0.2 (0.8)	0.3 (0.7)	0.0 (1.0)
C_4	0.7 (0.3)	0.2 (0.8)	0.3 (0.7)	0.0 (1.0)
C_5	0.2 (0.8)	0.7 (0.3)	0.1 (0.9)	0.0 (1.0)
C_6	0.0 (1.0)	0.0 (1.0)	0.0 (1.0)	0.9 (0.1)

Classification: Finding Content in a Candidate

Step 2 of 4

Determine the projection by finding the maximum weighted matching of the bipartite graph. Projection found: $R \sim \Pi_{C_4, C_2, C_1, C_6}(T)$:

	A_1	A_2	A_3	A_4
C_1			0.6 (0.4)	
C_2		0.9 (0.1)		
C_3				
C_4	0.7 (0.3)			
C_5				
C_6				0.9 (0.1)

Classification: Finding Content in a Candidate

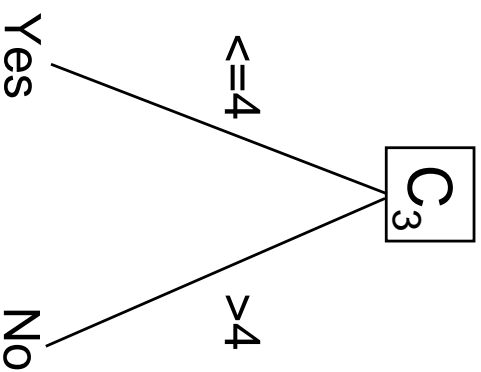
Step 3 of 4

Calculate the probability of each tuple being a contributor. Output is a set of labeled tuples. If $P(\text{Select}) \geq 0.5$, then label is "Yes."

C_1	C_2	C_3	C_4	C_5	C_6	$P(\text{Select})$
a	b	2	c	d	e	0.7
q	r	9	s	t	u	0.3
q	r	8	c	d	u	0.1
a	b	4	c	d	f	0.9

Classification: Finding Content in a Candidate**Step 4 of 4**

Determine the selection predicate by finding rules for the labeled tuples. Final expression $e_t = \Pi_{C_4, C_2, C_1, C_6}(\sigma_{C_3 \leq 4}(T))$



Validating the Approach

- We use *stratified cross-validation* for learning and classifying.
- For example, with a virtual database of two relations and three examples for each relation, we divide the mappings into three folds of equal content.

Fold	Virtual	Local
1	R_1	S_1, s_1, e_1
1	R_2	S_2, s_2, e_2
2	R_1	S_3, s_3, e_3
2	R_2	S_4, s_4, e_4
3	R_1	S_5, s_5, e_5
3	R_2	S_6, s_6, e_6

Measuring Performance

To measure performance, the outputs of Autoplex are regarded as four types of Boolean decisions:

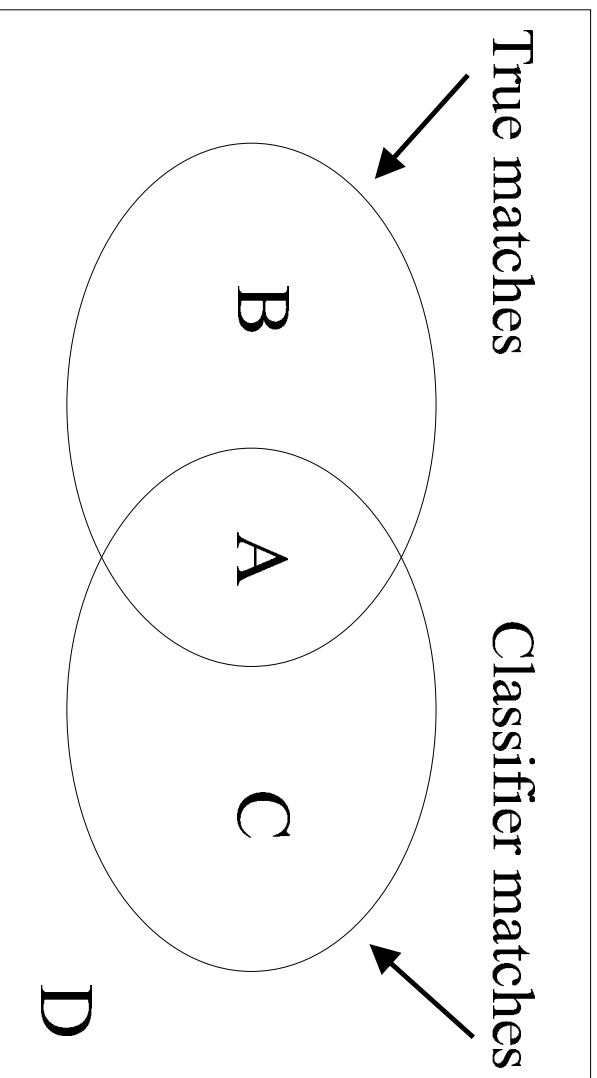
1. Column Mapping
2. Table Mapping
3. Tuple Partitioning
4. Tuple Selection

Measuring Performance (continued)

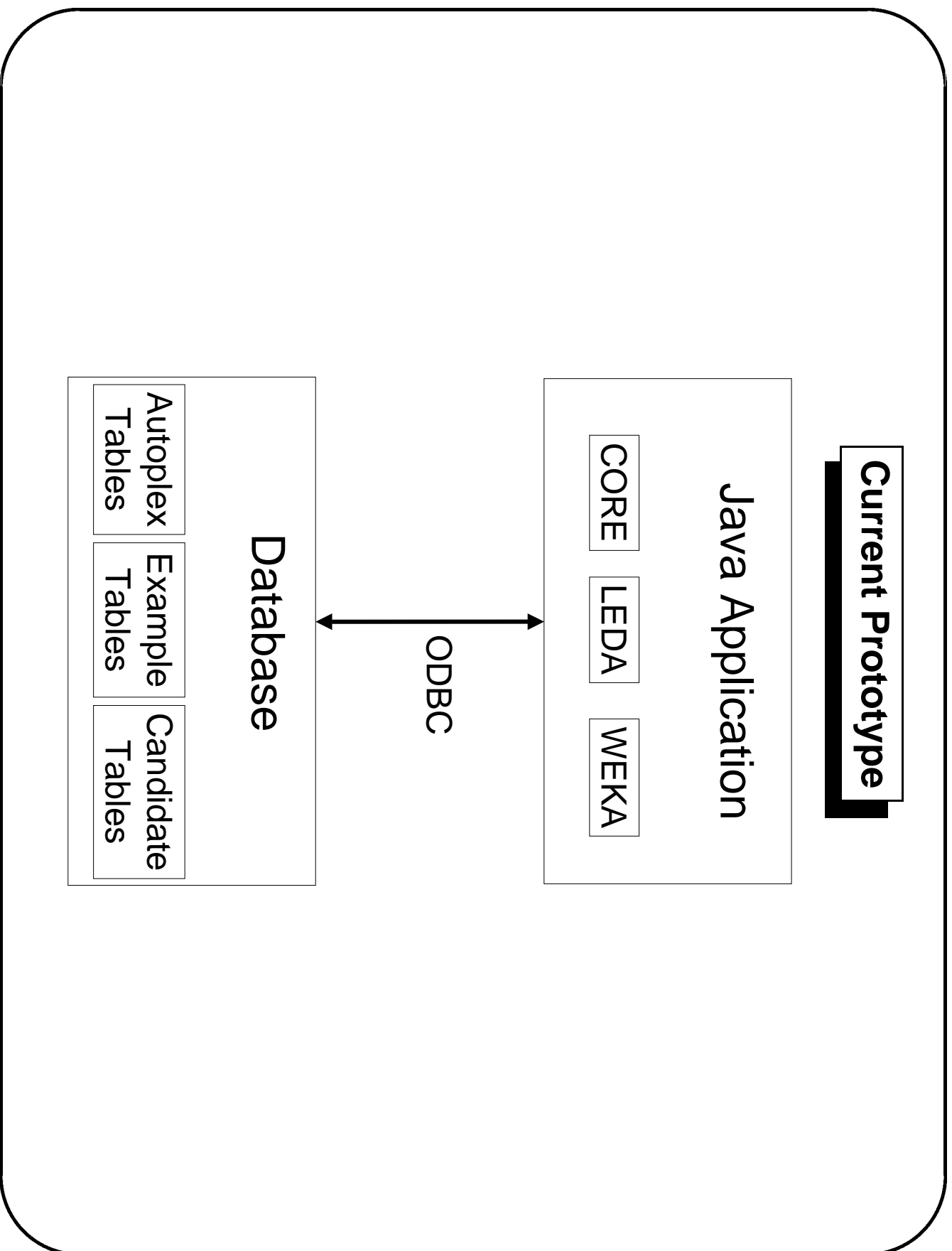
For each of these four decisions, the output falls into four disjoint categories:

- A:** Decision is *True* and the correct answer is *True*. Decisions in this partition are called **true positives**.
- B:** Decision is *False* and the correct answer is *True*. Decisions in this partition are called **false negatives**.
- C:** Decision is *True* and the correct answer is *False*. Decisions in this partition are called **false positives**.
- D:** Decision is *False* and the correct answer is *False*. Decisions in this partition are called **true negatives**.

Measuring Performance (continued)



$$\begin{aligned}
 \textit{Soundness} &= \frac{|A|}{|A|+|C|} \\
 \textit{Completeness} &= \frac{|A|}{|A|+|B|}
 \end{aligned}$$



Experiment

Defined a global scheme for computer retail information with the following relations:

1. Desktops = (Retailer, DesktopMaker, DesktopModel, DesktopCost, Availability)
2. Monitors = (Retailer, MonitorMaker, MonitorModel, PeripheralCost, Availability)
3. Printers = (Retailer, PrinterMaker, PrinterModel, PeripheralCost, Availability)

WWW Sources Used for Experiment

15 Source Tables, 21 Mappings, 1095 tuples

WWW Source	Virtual Database Relation
aberdeeninc.com	Printers
accessmicro.com	Monitors
beyond.com	Printers
buy.com	Desktops, Monitors
buybuddy.com	Printers
cnet.com	Monitors
compaq.com	Printers
compuplus.com	Monitors
cyberwarehouse.com	Desktops, Monitors
egghead.com	Desktops, Monitors
firstsource.com	Desktops
gateway.com	Monitors, Printers
maven.com	Printers
outpost.com	Desktops, Printers
pcstop.com	Desktops, Monitors

Empirical Results

Using stratified three-fold cross validation:

Category	A	B	C	D	Soundness	Completeness
Column Mapping	74	17	17	660	0.81	0.81
Table Mapping	18	3	0	24	1.00	0.86
Tuple Partitioning	969	60	117	612	0.89	0.94
Tuple Selection	967	62	104	625	0.90	0.94

Conclusions

- Machine Learning can be used to overcome scale-up barrier.
- Initial prototype and testing is promising.
- New paradigm: sources are discovered automatically and are *drafted* into the global scheme. Contrast with traditional method of waiting for someone to *volunteer* a new source.

Future Work

1. **Support More General Views:** Allow source and target views that involve *joins, unions, or transformations*.
2. **Use Intensional Information:** Extract intensional information from example sources and use this information in the discovery process (currently we use only extensional information).
3. **Assurance:** Combine the statistical performance of the discovery process with confidence measures of individual discoveries to produce a meaningful “pedigree” to be stored alongside the contribution.