DATA CLEANING

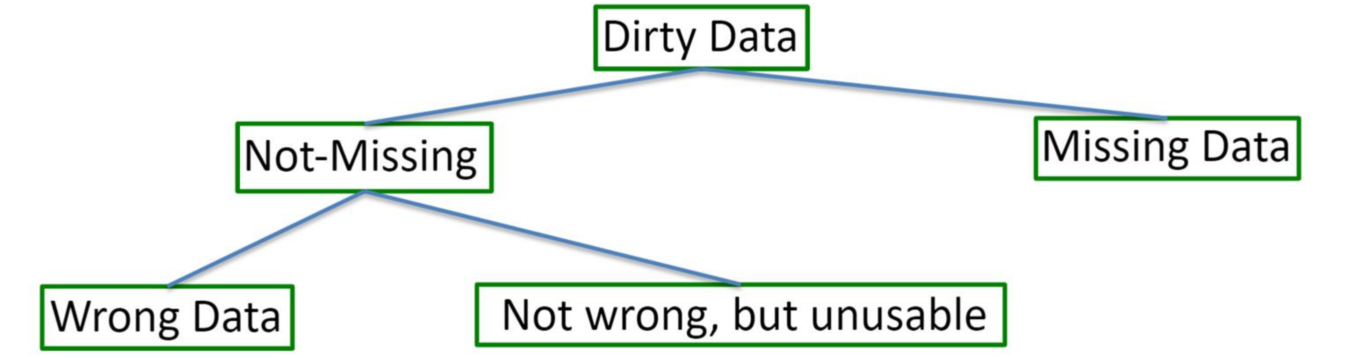
Minh Pham and Craig Knoblock

Outline

- Data Cleaning in Tabular Data
- Error Detection
- Data Transformation
- Imputing Missing Values
- OpenRefine

Data Cleaning Problem

- You have a data cleaning problem if the data doesn't look like WHAT YOU WANT IT TO BE.



artist	artistyear	artyear	credittext	dimensions	photo
FRANK STELLA	born 1936	1970	Gift of Alice Pratt Brown	120 x 600 inches	Oil on canvas
JASPER JOHNS	born 1930		Museum purchase with funds provided by; the Agnes Cullen Arnold Endowment Fund	75 x 50 inches	http://www.mfah.org/site_media/cache/e5/37/e537510de3215396c6c0336b17eab82b.jpg
GEORGE BELLOWS	1882 - 1925	1914	Gift of Mr. and Mrs. Meredith Long in memory of Agnes Cullen Arnold	38 x 30 inches	http://www.mfah.org/site_media/cache/72/7c/727c9b61b85b16d432a79031137e3d6b.jpg

Meaning of “WHAT YOU WANT IT TO BE ?”

Beer Name
21st Amendment IPA (2006)
Lower De Boom
Summer Solstice Cerveza Crema (2009)
Special Edition: Allies Win The War!
Passion Fruit Prussia
Rude Parrot IPA
"the Kimmie, the Yink and the Holy Gose"

The format of this name seems off. Is this an error ?

This look like a movie name. Is this an error ?

I never saw this beer name before. Is this an error ?

Definition of errors is totally based on user design/database schema

We can never solve the problem?

Beer Name	Beer Year
21st Amendment IPA (2006)	2006
Lower De Boom	2010
Summer Solstice Cerveza Crema (2009)	2009
Special Edition: Allies Win The War!	2006
Passion Fruit Prussia	2007
Rude Parrot IPA	2008
"the Kimmie, the Yink and the Holy Gose"	2011

Now is this an error ? **LIKELY**

What if I know this is an error ? **EVEN MORE**

Evidence in the tables can give you hints about data errors

Errors vs Outliers

GDP per capita
41 450
43 746
44 720
45 231

Is this an outlier ?

GDP per capita
41 450
43 746
44 720
45 231

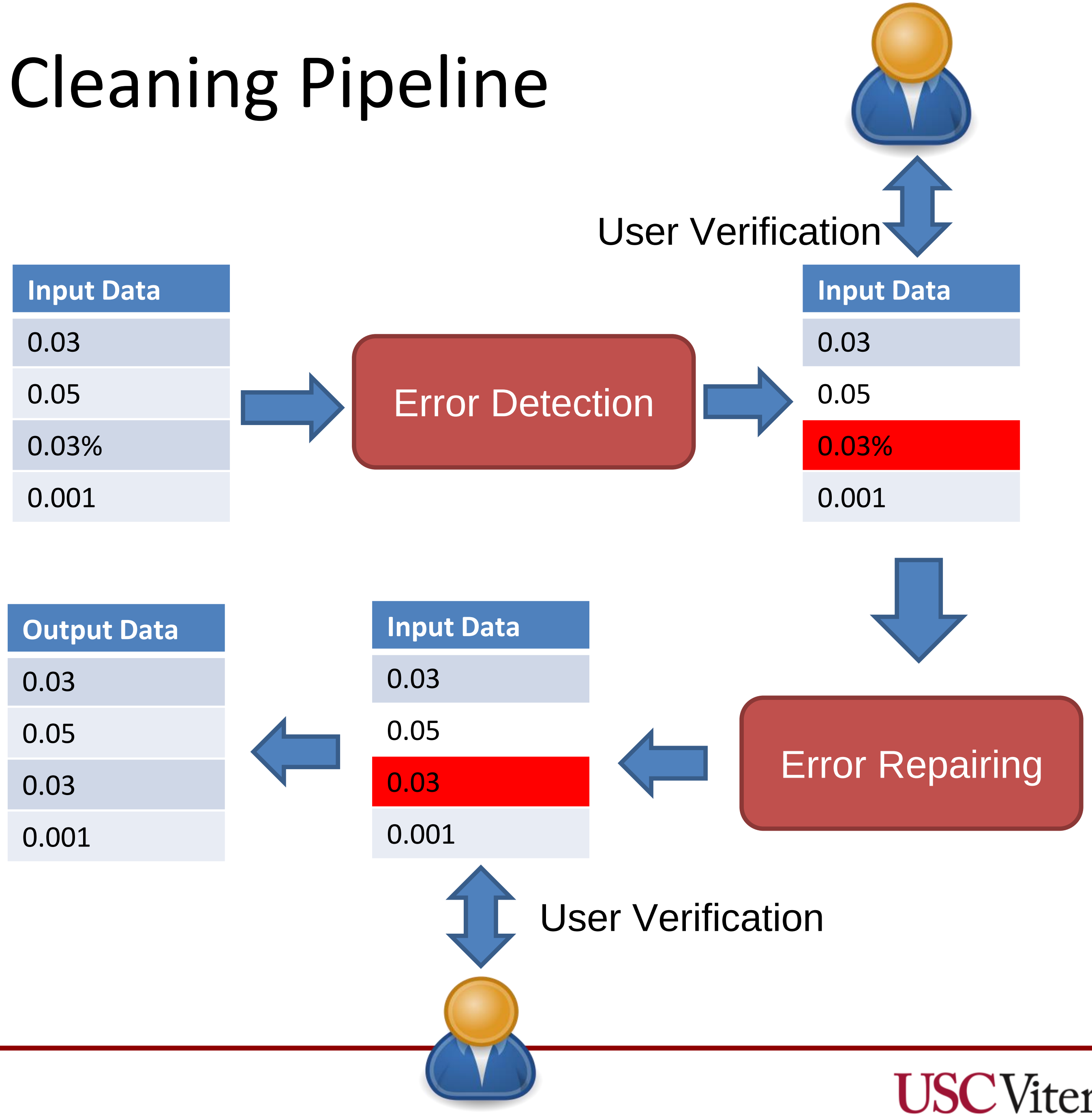
Is this an error ?

However, in a lot of cases, outliers are errors. So many outlier detection methods can be used to help detect errors.

Data Cleaning Pipeline

1. Detect errors in data

2. Repair errors in data



Data Cleaning Challenges

- Unknownness: You usually doesn't know how errors look like until you see it.

- Heterogeneous errors: There are many different type of errors

- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data.

Different Types of Data Errors

APY
0.05
0.02
0.1
0.05%
0.0100000001

Format Inconsistencies

State
California
Calofornia
Arizona
New York
California

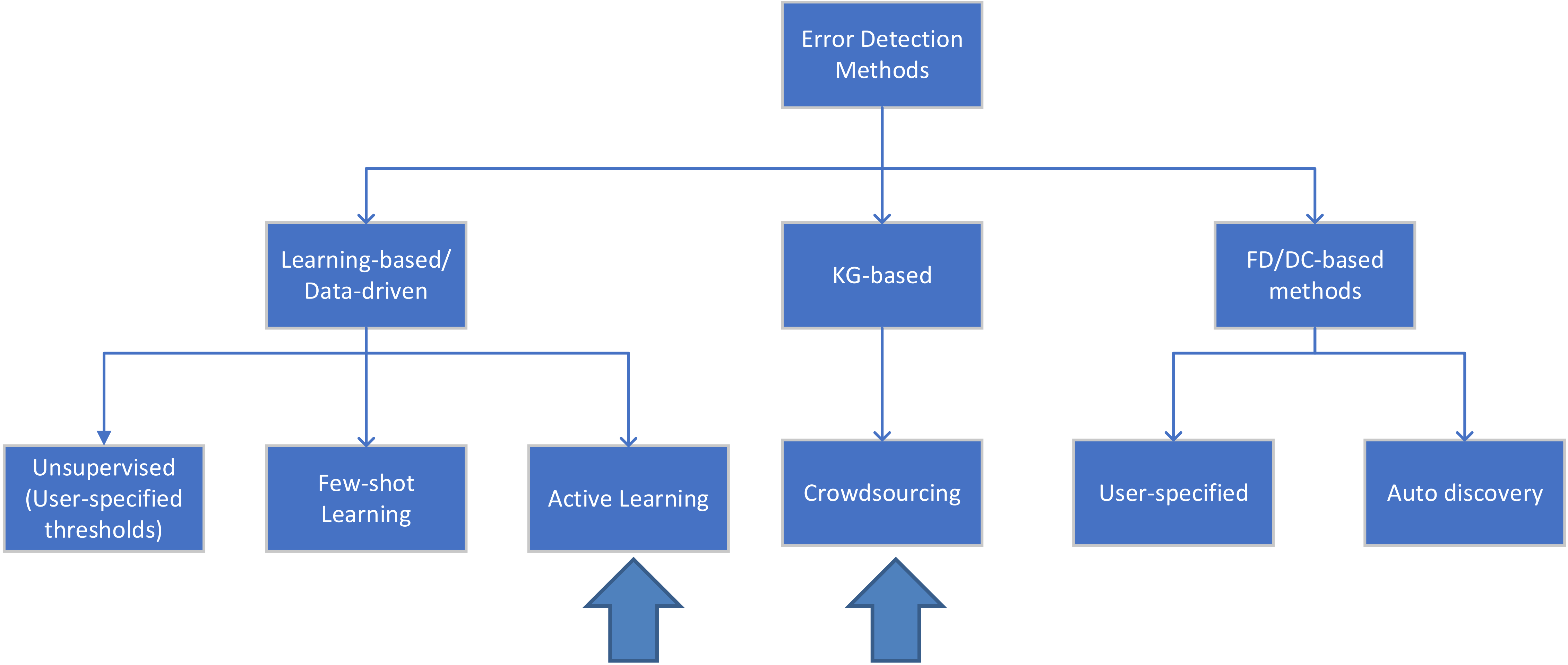
City
Los Angeles
Arizona
San Francisco
New York City
San Diego

Wrong values

City	State
Los Angeles	California
San Diego	Arizona
New York City	New York
Phoenix	Arizona
San Francisco	California

Violated attribute dependency

Error Detection Methods



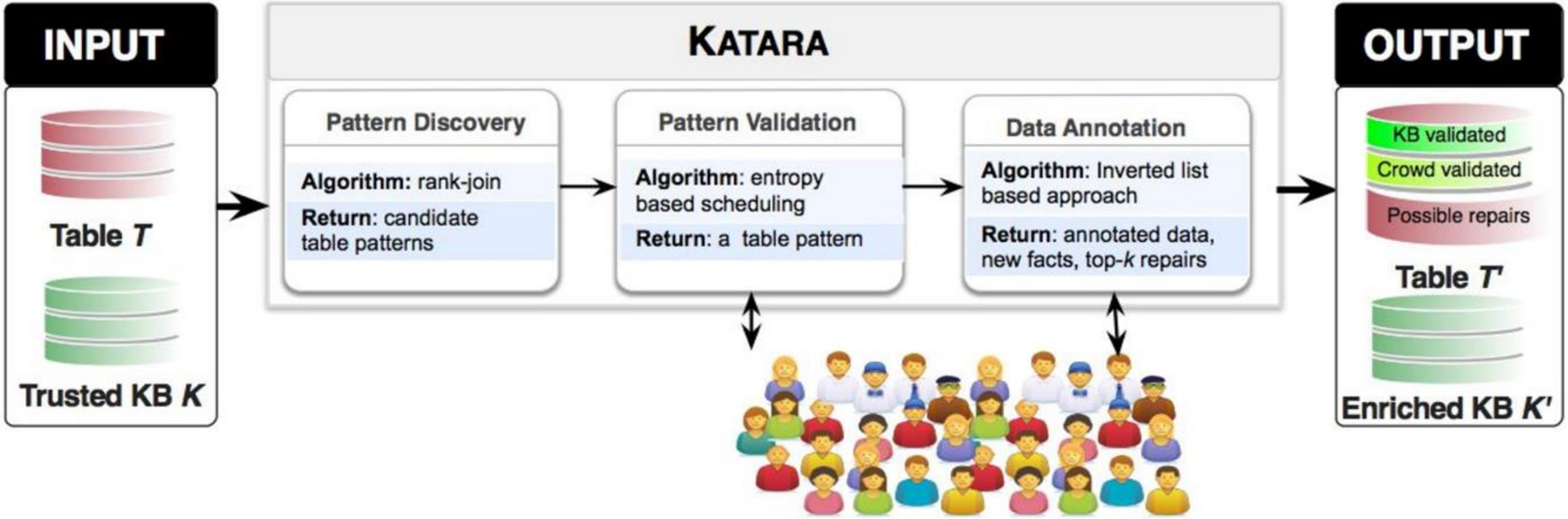
Slides from Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, Yin Ye.

KNOWLEDGE GRAPH BASED ERROR DETECTION

Challenges of KG-based Methods

- Unknownness: You usually doesn't know how errors look like until you see it
  - Anything that does not align with KG information is incorrect
- Heterogeneous errors: There are many different type of errors
  - Focus only on violated attribute dependency
- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data
  - Anything that does not align with KG information is incorrect

System Overview



What is KATARA?

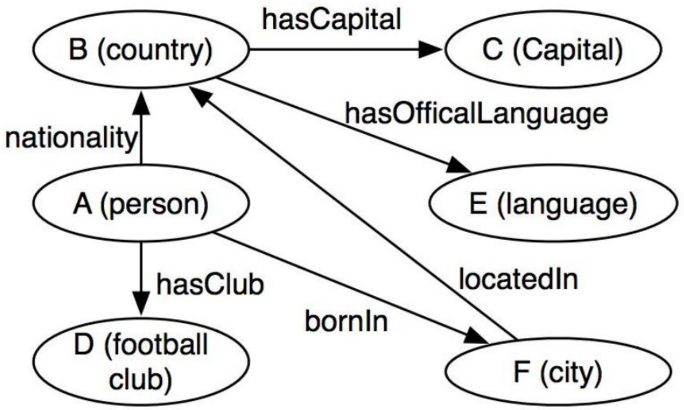
1. Table pattern definition and discovery (using KBs)
2. Table pattern validation via crowdsourcing
3. Data annotation
4. Repair recommendation

How can a knowledge graph help us clean data ?

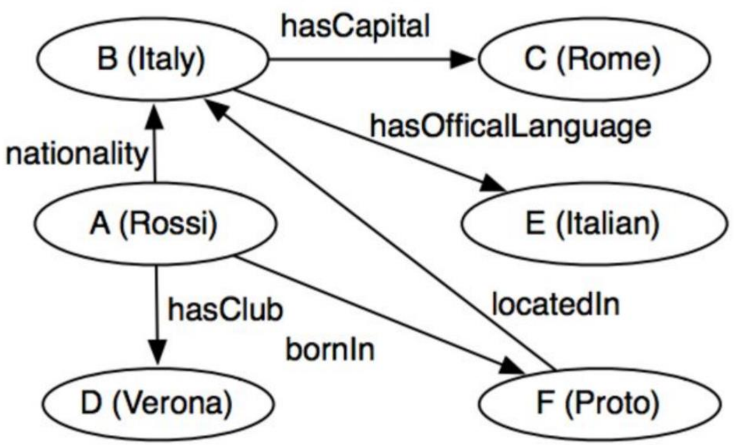
Table pattern semantics

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	S. Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

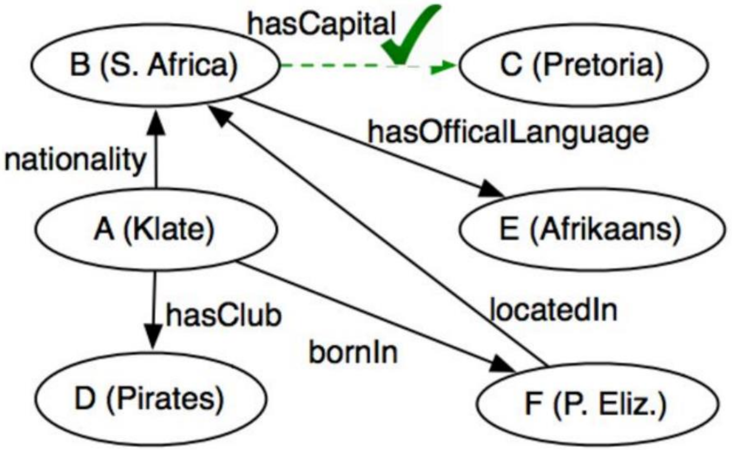
Semantic Labeling
+
Source Modeling



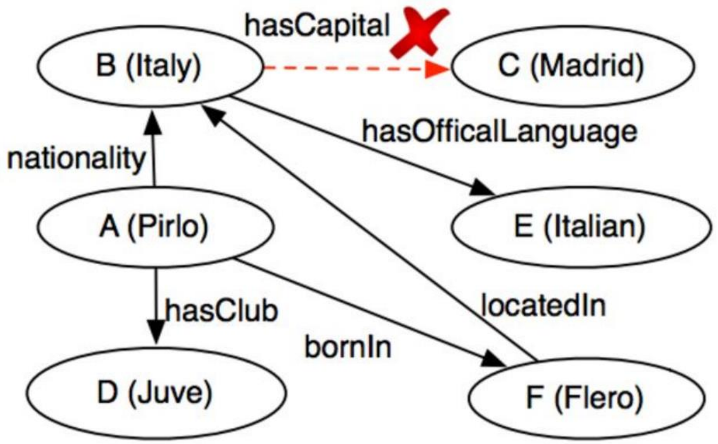
KGs and source models: different situations



Full KB Coverage



Does S. Africa hasCapital “Pretoria?”



Does Italy hasCapital “Madrid?”

Partial KB Coverage

KGs and source models: different situations

- What are the relationships between Rossi and 1.78?

- Not covered by the KB => Not covered by the system

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	S. Africa	Pretoria	Pirates	Afrikaans	P. Eliz	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

KATARA: core component

How do we actually extract knowledge from KGs?

KATARA: SPARQL queries

$$
\begin{array}{ll}
Q_{\text{types}} & \textbf{select } ?c_i \\
& \textbf{where } \{?x_i \text{ rdfs:label } t[A_i], \\
& \qquad ?x_i \text{ rdfs:type/rdfs:subClassOf}{*} \ ?c_i\}
\end{array}
$$

Get types and supertypes of resources with value t[A_i]

KATARA: SPARQL queries

$$
\begin{array}{ll}
Q^1_{\text{rels}} & \textbf{select } ?P_{ij} \\
 & \textbf{where } \{?x_i \text{ rdfs:label } t[A_i],\ ?x_j \text{ rdfs:label } t[A_j], \\
 & \qquad ?x_i\ ?P_{ij}/\text{rdfs:subPropertyOf}*\ ?x_j\} \\
Q^2_{\text{rels}} & \textbf{select } ?P_{ij} \\
 & \textbf{where } \{?x_i \text{ rdfs:label } t[A_i], \\
 & \qquad ?x_i\ ?P_{ij}/\text{rdfs:subPropertyOf}*\ t[A_j]\}
\end{array}
$$

Q1: get relationships where both attributes are resources

Q2: get relationships between one resource and one literal

Evaluating KB types (T) for attributes (A)

$$\text{tf}(T_i, t[A_i]) = \begin{cases} 0 & \text{if } t[A_i] \text{ is not of Type } T_i \\ \frac{1}{\log(\text{Number of Entities of Type } T_i)} & \text{otherwise} \end{cases}$$

$$\text{idf}(T_i, t[A_i]) = \begin{cases} 0 & \text{if } t[A_i] \text{ has no type} \\ \log \frac{\text{Number of Types in } \mathcal{K}}{\text{Number of Types of } t[A_i]} & \text{otherwise} \end{cases}$$

Intuitively, the less the number of types t[Ai] has, the more contribution t[Ai] makes.

$$\text{tf-idf}(T_i, A_i) = \sum_{t \in \mathcal{T}} \text{tf-idf}(T_i, t[A_i])$$

Evaluating KB types (T) for attributes (A)

$$\text{tf}(T_i, t[A_i]) = \begin{cases} 0 & \text{if } t[A_i] \text{ is not of Type } T_i \\ \frac{1}{\log(\text{Number of Entities of Type } T_i)} & \text{otherwise} \end{cases}$$

$$\text{idf}(T_i, t[A_i]) = \begin{cases} 0 & \text{if } t[A_i] \text{ has no type} \\ \log \frac{\text{Number of Types in } \mathcal{K}}{\text{Number of Types of } t[A_i]} & \text{otherwise} \end{cases}$$

$$\text{tf-idf}(T_i, A_i) = \sum_{t \in \mathcal{T}} \text{tf-idf}(T_i, t[A_i])$$

Column A
Microsoft
Apple

Which cell is more important ?

Evaluating possible column relationships

prob. of an entity being of type *T* and appearing as subject of *P*

$$\mathsf{PMI}_{\mathsf{sub}}(T, P) = \log \frac{\mathsf{Pr}_{\mathsf{sub}}(P \cap T)}{\mathsf{Pr}_{\mathsf{sub}}(P)\mathsf{Pr}(T)}$$

prob. of any entity appearing in the subject of property *P*

prob. of any entity being of type *T*

Quantify the “compatibility” between a type T and relationship P

Evaluating possible column relationships

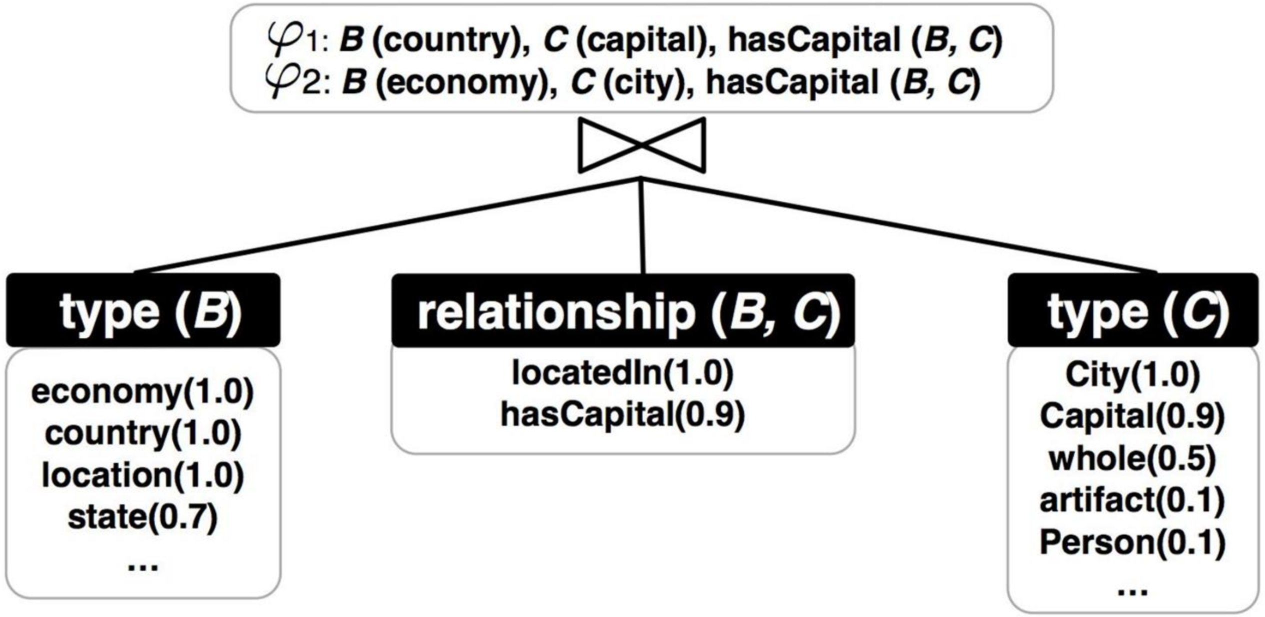
- From PMI we get a measure of semantic coherence

$$\mathsf{subSC}(\mathsf{economy}, \mathsf{hasCapital}) = 0.84$$
$$\mathsf{subSC}(\mathsf{country}, \mathsf{hasCapital}) = 0.86$$
$$\mathsf{objSC}(\mathsf{city}, \mathsf{hasCapital}) = 0.69$$
$$\mathsf{objSC}(\mathsf{capital}, \mathsf{hasCapital}) = 0.83$$

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	S. Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

Generate top-k patterns

$$\text{score}(\varphi) = \Sigma_{i=0}^{m} \text{tf-idf}(T_i, A_i) + \Sigma_{ij} \text{tf-idf}(P_{ij}, A_i, A_j) + \Sigma_{ij}(\text{subSC}(T_i, P_{ij}) + \text{objSC}(T_j, P_{ij}))$$



Top-k to final answer

- Crowdsourcing – asking the crowd for help
- Convert table patterns into questions

Q_1 : *What is the most accurate type of the highlighted column?*

(A, **B**, C, D, E, F, ...)

(Rossi, **Italy**, Rome, Verona, Italian, Proto, ...)

(Pirlo, **Italy**, Madrid, Juve, Italian, Flero,, ...)

○ country ○ economy
○ state ○ none of the above

Q_2 : *What is the most accurate relationship for* highlighted columns (A, **B, C**, D, E, F, ...)

(Rossi, **Italy, Rome**, Verona, Italian, Proto, ...)

(Pirlo, **Italy, Madrid**, Juve, Italian, Flero, ...)

○ B hasCapital C ○ C locatedIn B ○ none of the above

Pattern Validation

- Query the crowd for validation

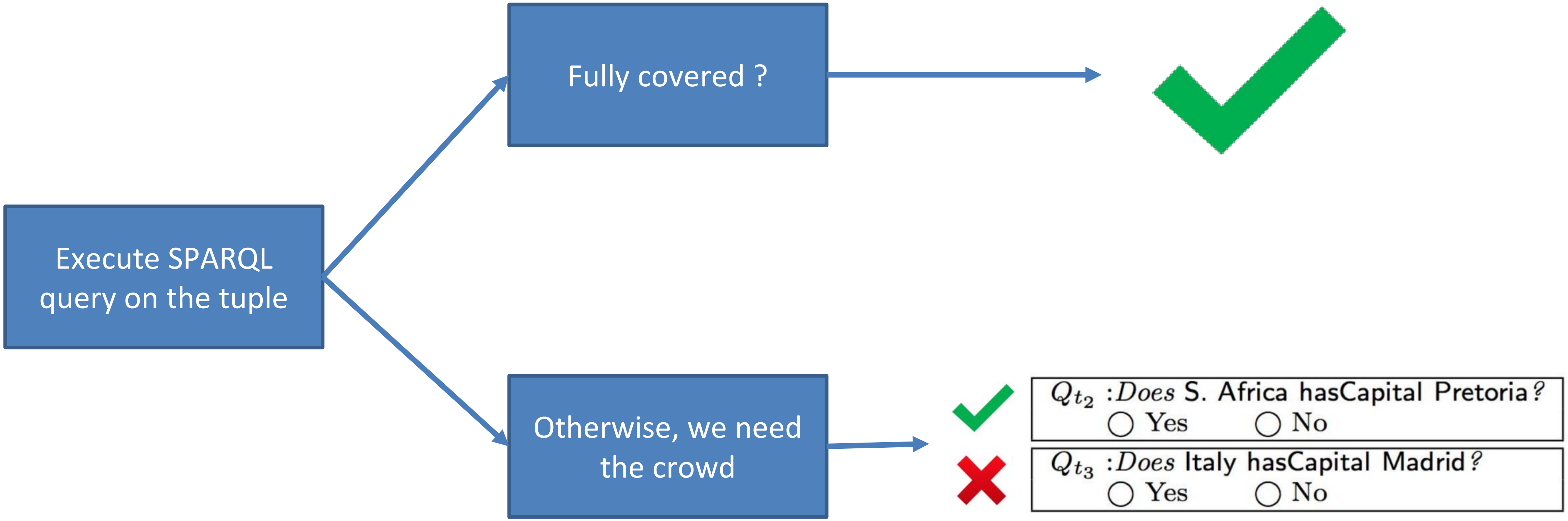
	type (B)	type (C)	$P(B, C)$	score	prob
φ_1	country	capital	hasCapital	2.8	0.35
φ_2	economy	capital	hasCapital	2	0.25
φ_3	country	city	locatedIn	2	0.25
φ_4	country	capital	locatedIn	0.8	0.1
φ_5	state	capital	hasCapital	0.4	0.05

Q_1 : *What is the most accurate type of the highlighted column?*
(A, B, C, D, E, F, ...)
(Rossi, Italy, Rome, Verona, Italian, Proto, ...)
(Pirlo, Italy, Madrid, Juve, Italian, Flero,, ...)
◯ country ◯ economy
◯ state ◯ none of the above

- Remove tuples that violate validation, and repeat above until left with one pattern

	type (B)	type (C)	$P(B, C)$	prob
φ_1	country	capital	hasCapital	0.5
φ_3	country	city	locatedIn	0.35
φ_4	country	capital	locatedIn	0.15

Summary


Execute SPARQL query on the tuple
Fully covered ?
Otherwise, we need the crowd
Q_{t_2} : Does S. Africa hasCapital Pretoria?
Yes No
Q_{t_3} : Does Italy hasCapital Madrid?
Yes No

MACHINE LEARNING BASED ERROR DETECTION

Challenges of Machine Learning Methods

- Unknownness: You usually doesn't know how errors look like until you see it
  - Educated "guessing" + user verification
- Heterogeneous errors: There are many different type of errors
  - Different types of features and/or different of sub model

Correlated

- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data
  - Few-shot + Semi-supervised learning techniques

Challenges of Machine Learning Methods

- Unknownness: You usually doesn't know how errors look like until you see it
  - Educated "guessing" + user verification

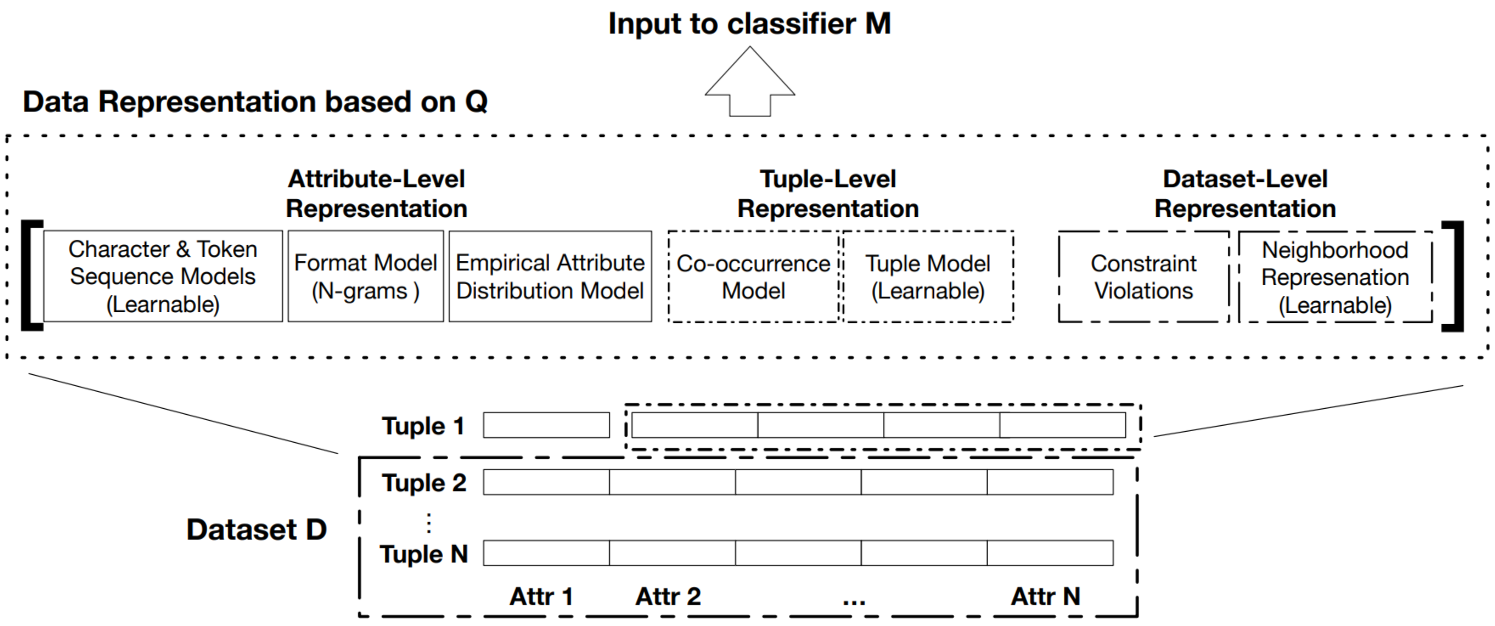
Correlated

- Heterogeneous errors: There are many different type of errors
  - Different types of features and/or different of sub model

- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data
  - Few-shot + Semi-supervised learning techniques

How to cover a variety of errors?

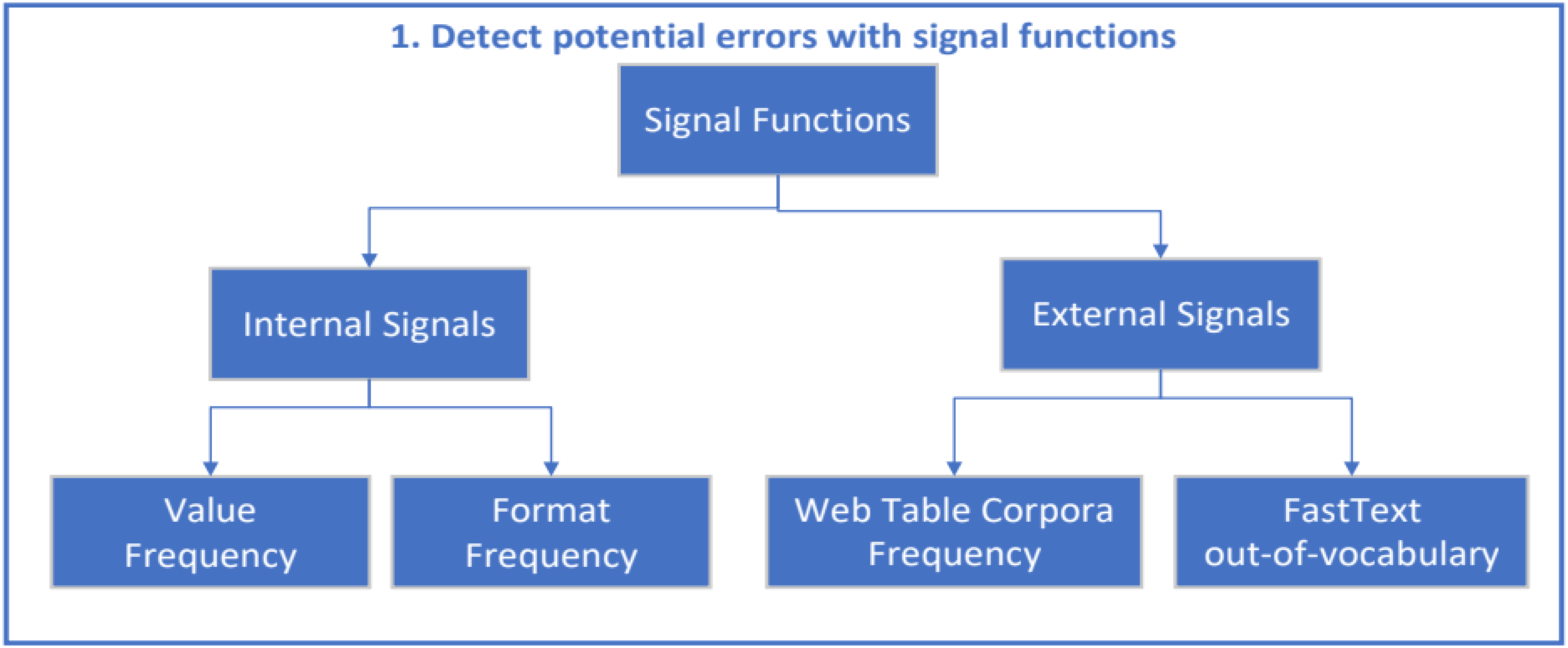
- HoloDetect (Heidari et al, 2019): Variety of features to represent different aspect of data



(A) Schematic Diagram of Representation Model Q

How to cover a variety of errors?

- SPADE: Variety of sub-models:



Challenges of Machine Learning Methods

- Unknownness: You usually doesn't know how errors look like until you see it
  - Educated "guessing" + user verification
- Heterogeneous errors: There are many different type of errors
  - Different types of features and/or different of sub model

Correlated

- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data
  - Few-shot + Semi-supervised learning techniques

How to make educated guess ?

- SPADE

Signal Function

Detect potential errors

Input Data
0.03
0.05
0.03%
0.001

Align

Input Data
0.03
0.05
0.03%
0.001

User Labeling

Increase Signal Function Weight

Signal Function 4

Detect potential errors

Input Data
1.3
2.4
-0.5
1.5

Not Align

Input Data
1.3
2.4
-0.5
1.5

User Labeling

Decrease Signal Function Weight

Challenges of Machine Learning Methods

- Unknownness: You usually doesn't know how errors look like until you see it
  - Educated "guessing" + user verification
- Heterogeneous errors: There are many different type of errors
  - Different types of features and/or different of sub model

Correlated

- Rarity and class imbalance: Errors are typically rare, contrasting to normal instances that often account a large portion of the data
  - Few-shot + Semi-supervised learning techniques

Error Detection as a Few-shot Learning Problem

Labeled Errors	Corrected values
5.6%	5.6
0.500006	0.5
…	…

Learn
noisy channel

Transformation	Probability
Add % to the end of string	0.5
Add "00006" to the end of string	0.5
…	….

User labeled a small set of data

Apply noisy channel on normal values

Labeled Normal values
3.4
2.3
0.7
0.8

Artificial errors
300006.4
2.%3
0.7%
0.000068

Semi-Supervised Error Detection

- Raha

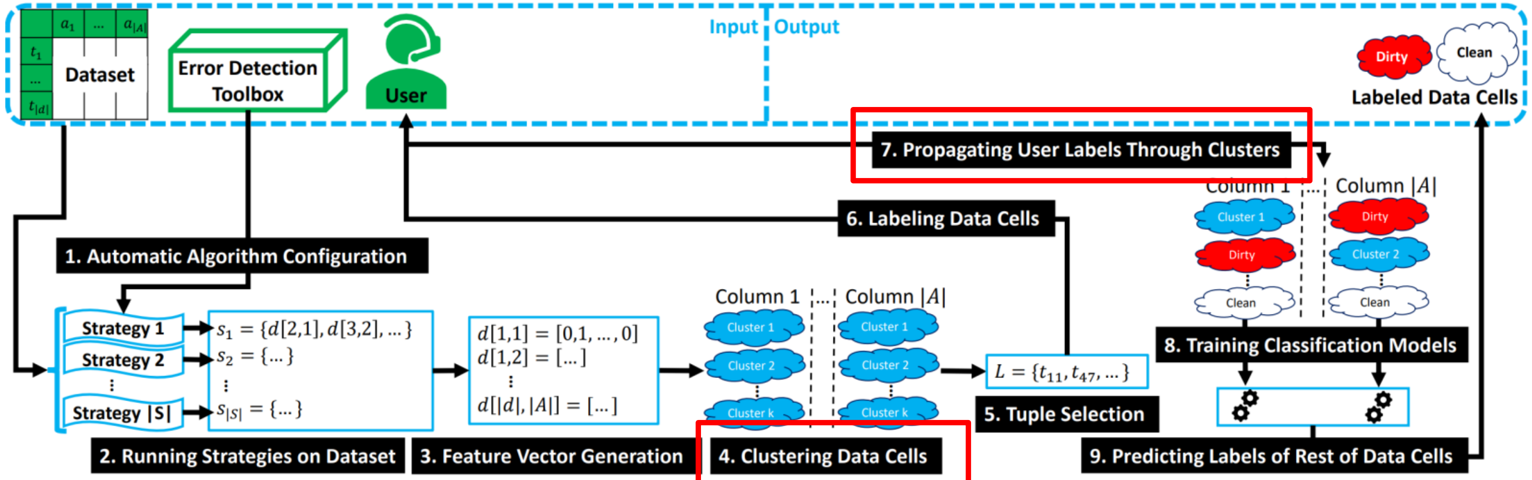
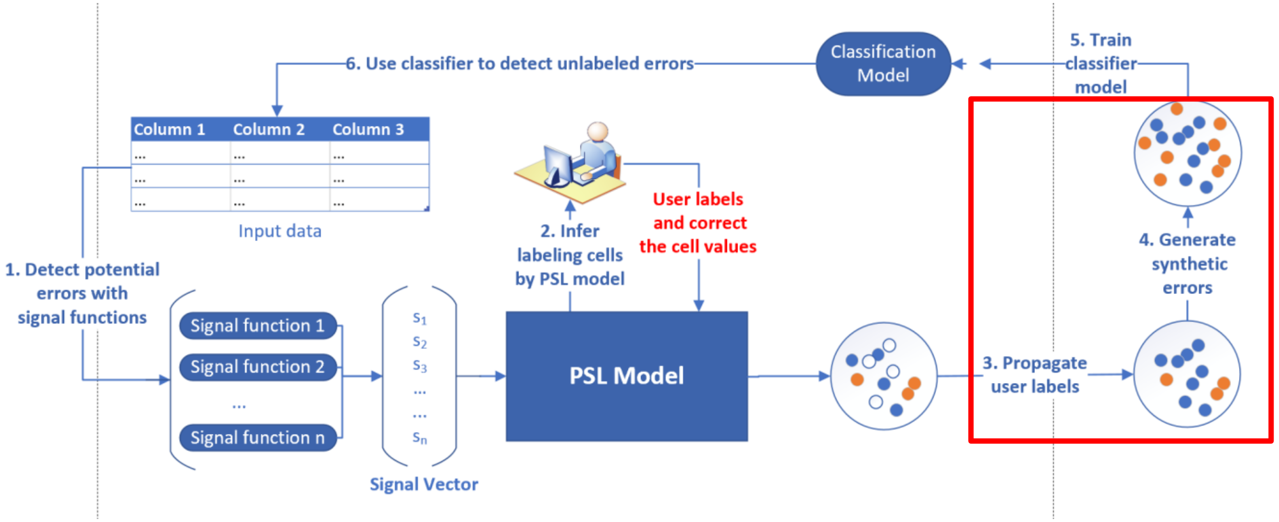


Figure 1: The workflow of Raha.

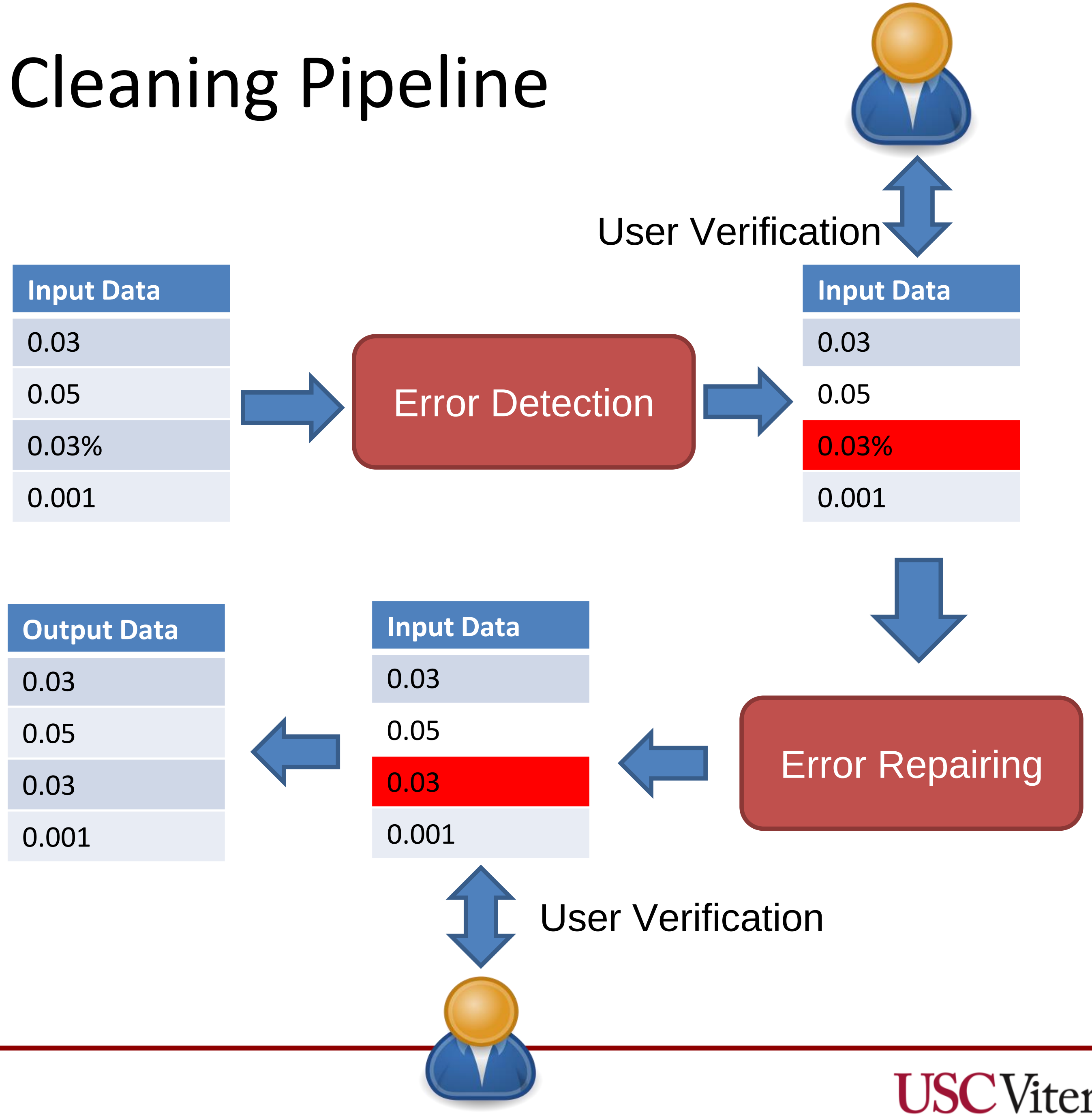
- SPADE



Data Cleaning Pipeline

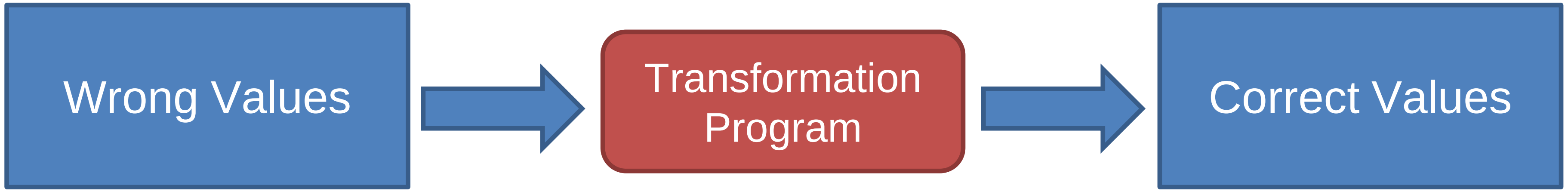
1. Detect errors in data

2. Repair errors in data



Data Transformation

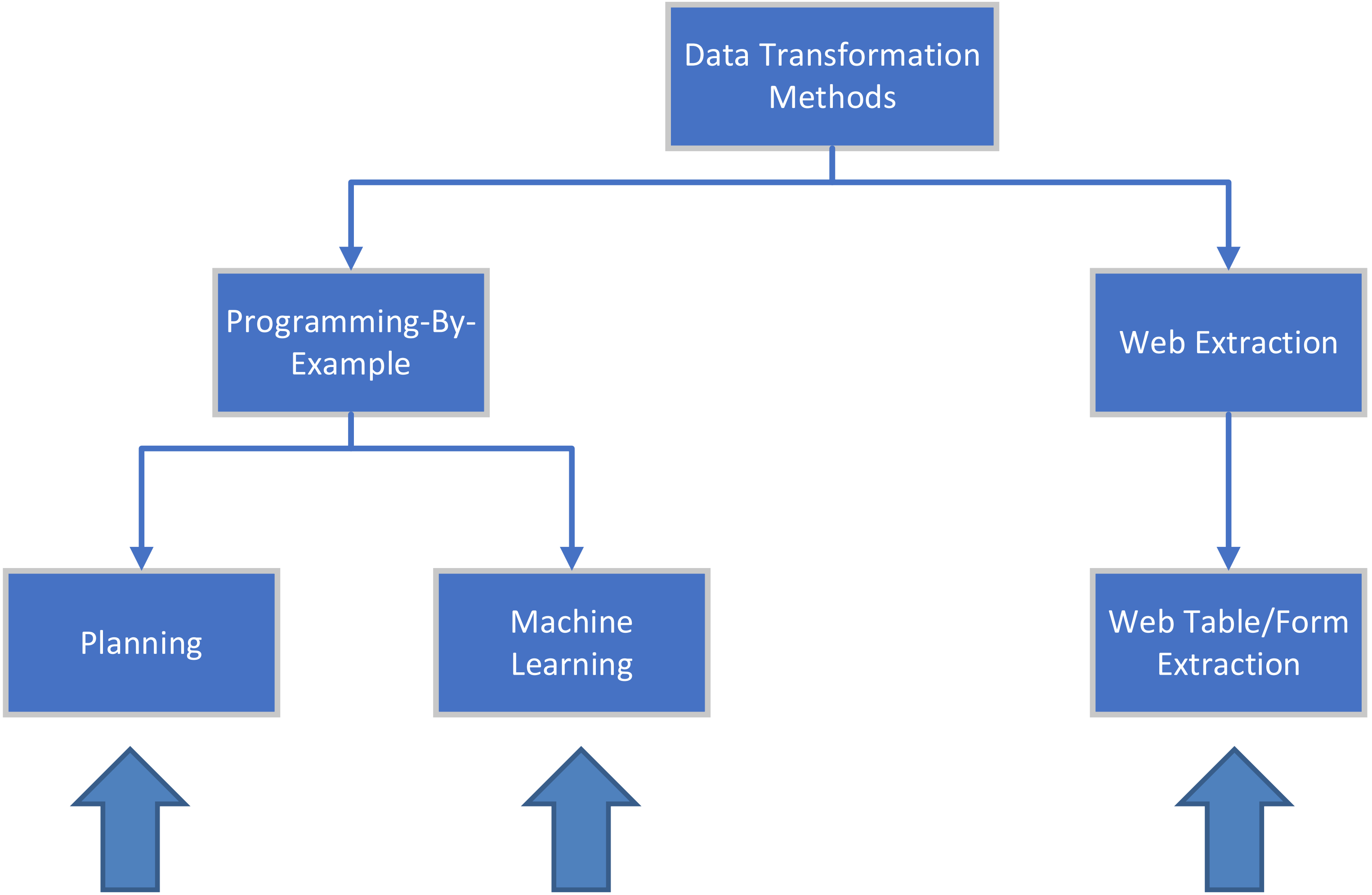
- Why data transformation ?



Input Data
0.03
0.05
0.03%
0.001

Output Data
0.03
0.05
0.03
0.001

Data Transformation Methods

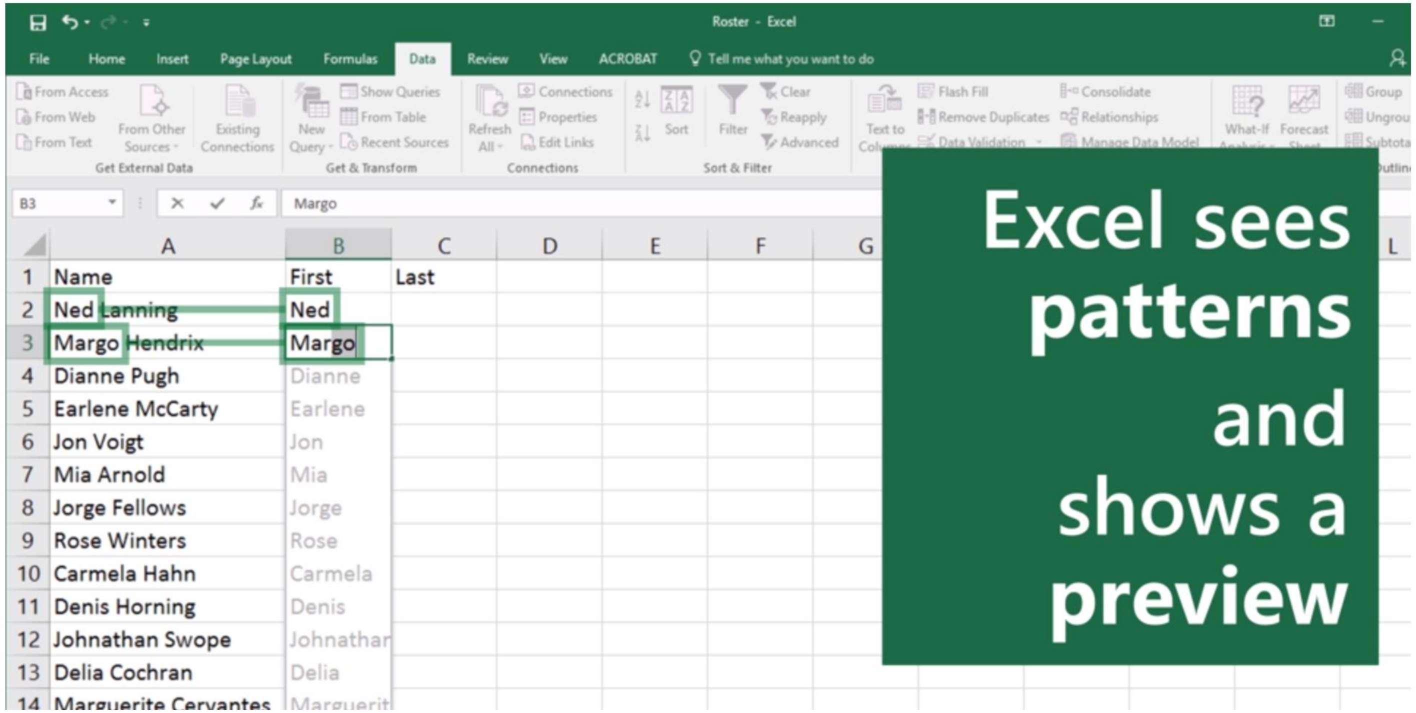


PLANNING FOR PROGRAMMING-BY-EXAMPLE TRANSFORMATION

What is Programming-By-Example Transformation ?

- PBE: Teaching systems new behavior using concrete examples.

- PBE Transformation: Systems learn transformation program as user input examples



PBE Architecture

Inductive Spec

Program

Search Algorithm

DSL

Inverse semantics of operators
for problem reduction

Transformation Program

Transformation Programs = Multiple Branch Programs

Branch Program = Substring Programs + Constant Programs

Substring Programs contains 2 Positions.

BNK: whitespace
NUM([0-9]+): 98
UWRD([A-Z]): I
LWRD([a-z]+): mage
WORD([a-zA-Z]+): Image

$Transform(value)$

Conditional statement

$$\boxed{\text{switch (classify(value)) :}}$$

case format$_1$:

Branch transformation program

$$\boxed{\begin{array}{l} \text{pos}_1 = value.indexOf(BNK,\ \text{NUM, -1}) \\ \text{pos}_2 = value.indexOf(NUM,\ \text{BNK, 2}) \\ \text{output=value.substr(pos}_1,\ \text{pos}_2) \end{array}}$$

case format$_2$:

Branch transformation program

$$\boxed{\begin{array}{l} \text{pos}_3 = value.indexOf(\text{"|"},\ \text{NUM, 2}) \\ \text{pos}_4 = value.indexOf(NUM,\ BNK,\ \text{-1}) \\ \text{output=value.substr(pos}_3,\ \text{pos}_4) \end{array}}$$

return output

9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE => 19.5

Transformation Program

BNK: whitespace
NUM([0-9]+): 98
UWRD([A-Z]): I
LWRD([a-z]+): mage
WORD([a-zA-Z]+): Image

9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE

19.5

$Transform(value)$

Conditional statement

switch (classify(value)) :

case format_1 :

Branch transformation program

$pos_1 = value.indexOf(BNK, \text{NUM}, \text{-1})$

$pos_2 = value.indexOf(NUM, \text{BNK}, 2)$

$\text{output=value.substr}(pos_1, \ pos_2)$

case format_2 :

Branch transformation program

$pos_3 = value.indexOf(\text{"|"}, \text{NUM}, 2)$

$pos_4 = value.indexOf(NUM, BNK, \text{-1})$

$\text{output=value.substr}(pos_3, \ pos_4)$

return output

Creating Hypothesis Spaces

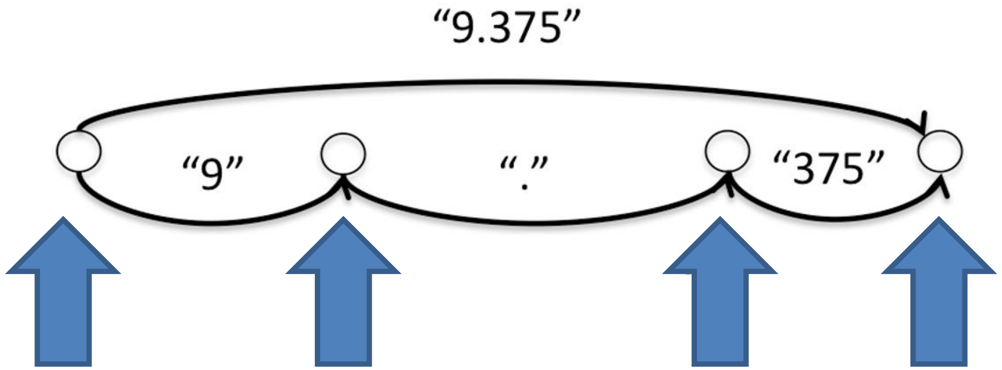
- Create traces

__Traces__: A trace here defines how the output string is constructed from a specific set of substrings from the input string.

Original: 5.25 in HIGH x 9.375 in WIDE

Target: 9.375

- Derive hypothesis spaces



Learning Conditional Statements

- Cluster-as-you-go: depends on transformations => split when there is no valid program.

R_1	5.25 in HIGH x 9.375 in WIDE	9.375	Cluster1-format$_1$
R_2	20 in HIGH x 24 in WIDE	24	
R_4	Image: 20.5 in. HIGH x 17.5 in. WIDE	17.5	
R_3	9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	19.5	Cluster2-format$_2$

- Learn a multiclass classifier
  - Recognize the format/regex of the inputs

R_5	Image: 20.5 in. HIGH x 17.5 in. WIDE	format$_1$
R_6	12 in|14 in HIGH x 16 in|18 in WIDE	format$_2$

Ranking

Synthesize multiple programs & rank them.

Basic ranking scheme

- Define a partial order over program expressions.
  - Prefer shorter programs.
  - Prefer programs with fewer constants.

Machine-learning based ranking

- Score using a weighted combination of program features.
  - Weights are learned using training data.

Issues with PBE approach

Telephone numbers	User examples
(213) 713-5203	213-713-5203
(213) 713-3475	
385-238-5592	
(385) 724-2345	
213.124.9963	
+1 385-554-0675	
385-218-4325	
…..	

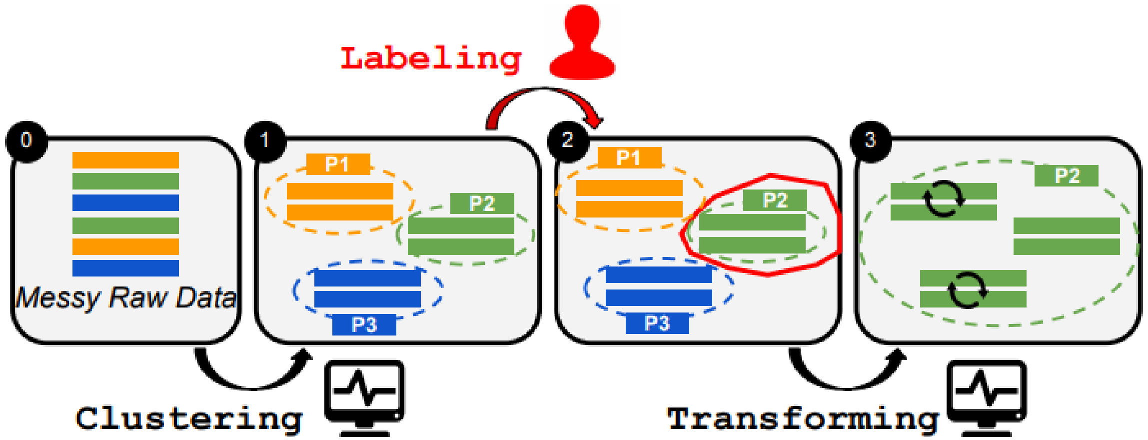
Telephone numbers	Output Value
(213) 713-5203	213-713-5203
(213) 713-3475	213-713-3475
385-238-5592	385-238-5592
(385) 724-2345	385-724-2345
213.124.9963	213-124-9963
+1 385-554-0675	**1-385-554**
385-218-4325	385-218-4325
…..	

Too Many Records for PBE


```
"26"" H x 24"" W x 12.5"" D","26"
"74"" H x 31.5"" W","74"
"10"" H x 8"" W","10"
"18"" H x 8.5"" W x 4"" D","18"
"33 x 25""","33"
"49.5 x 41""","49.5"
"31.75 x 26.25""","31.75"
"Framed at 21.75"" H x 24.25"" W","21.75"
"10"" H x 7.25"" W","10"
"11.75"" H  x 8.75"" W","11.75"
"10"" H x 7.25"" W","10"
"31"" H x 25"" W","31"
"32.5"" H x 15"" W x 10"" D","32.5"
"32.25"" H x 15"" W x 10""D","32.25"
"75"" H x 20""W x 12"" D","75"
"33"" H x 35.25"" W","33"
"19.5 x 25 x 2.25""","19.5"
"49"" x 5.5""","49"
"14"" H x 11"" W","14"
"31"" x 6.5""","31"
"36"" H x 32"" W","36"
"25"" H x 18"" W x 22"" D","25"
"16.25"" x 5.5"" x 5""","16.25"
"30 x 46""","30"
"20.5"" x 16""","20.5"
"36"" H x 24"" W","36"
"12"" H x 9"" W","12"
"41"" x 14.5"" x 4.5""","41"
"21.5"" x 29"" x 0.75""","21.5"
"10.75"" x 19.75"" x 1""","10.75"
"12.75"" x 10.75"" x 1""","12.75"
"49.75"" x 23.5"" x 1.75""","49.75"
"14.75"" H x 11"" W","14.75"
"14.75"" H x 11"" W","14.75"
"14.75"" H x 11"" W","14.75"
"14.75"" H x 11"" W","14.75"
"14.75"" H x 11"" W","14.75"
"14.75"" H x 11"" W","14.75"
"15.25"" H x 12"" W x 1"" D","15.25"
"21.5"" H x 27"" W x 1"" D","21.5"
"77"" H x 9""W  x 7"" D","77"
"10"" H x 8"" W","10"
"10"" H x 8"" W","10"
"70"" H x 11"" W x 3"" D","70"
"3.375"" H x 2.0125"" W","3.375"
"29.25 x 13 x 8.5""","29.25"
```

Minimizing User Effort: CLX

- CLX: Cluster-Label-Transform (Jin et al, 2019)
  - Clustering before Labeling instead of Clustering based on Labeling



Minimizing User Effort: Active Learning

Entire dataset

Raw
10“ H x 8” W
H: 58 x W:25”
12”H x 9”W
11”H x 6”
…
30 x 46”

Random Sampling

Sampled records

Raw	Transformed
10“ H x 8” W	10
11”H x 6”	11
…	…
30 x 46”	30 x 46

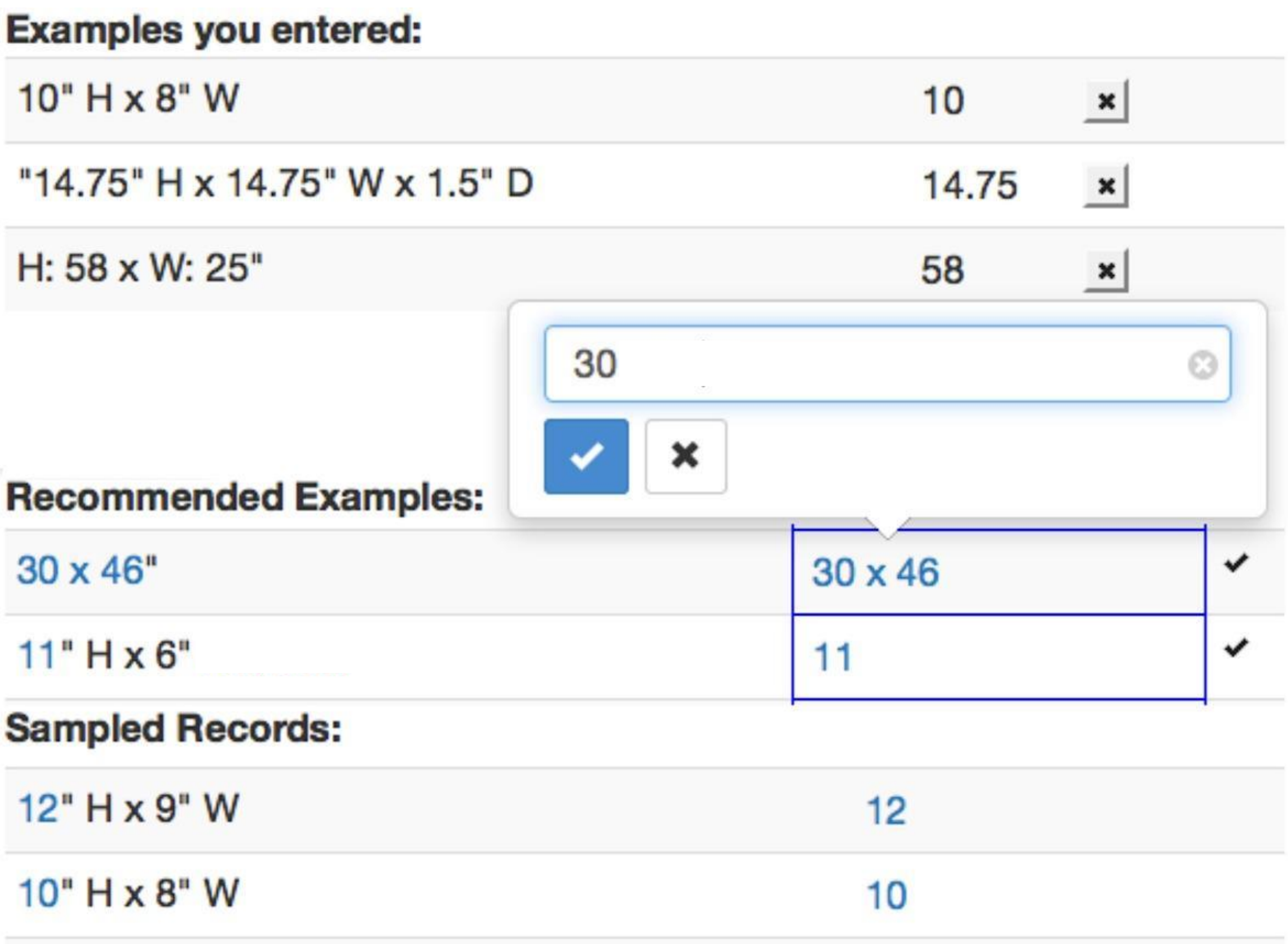
Verifying records

Raw	Transformed
11”H x 6”	11
30 x 46”	30 x 46
…	…

Sorting and color-coding

Raw	Transformed
30 x 46”	30 x 46
11”H x 6”	11
…	…

Learning from users' feedback



Color-coded Output

Entire dataset

Raw	Transformed
10“ H x 8” W	10
H: 58 x W:25”	58
12”H x 9”W	12
11”H x 6”	11
…	…
30 x 46”	30 x 46

Random Sampling

Sampled records

Raw	Transformed
10“ H x 8” W	10
11”H x 6”	11
…	…
30 x 46”	30 x 46

Verifying records: Active Learning

Raw	Transformed
11”H x 6”	11
30 x 46”	30 x 46
…	…

Sorting and color-coding

Raw	Transformed
30 x 46”	30 x 46
11”H x 6”	11
…	…

Verifying Records

- Recommend records causing runtime errors
  - Records cause the program exit abnormally

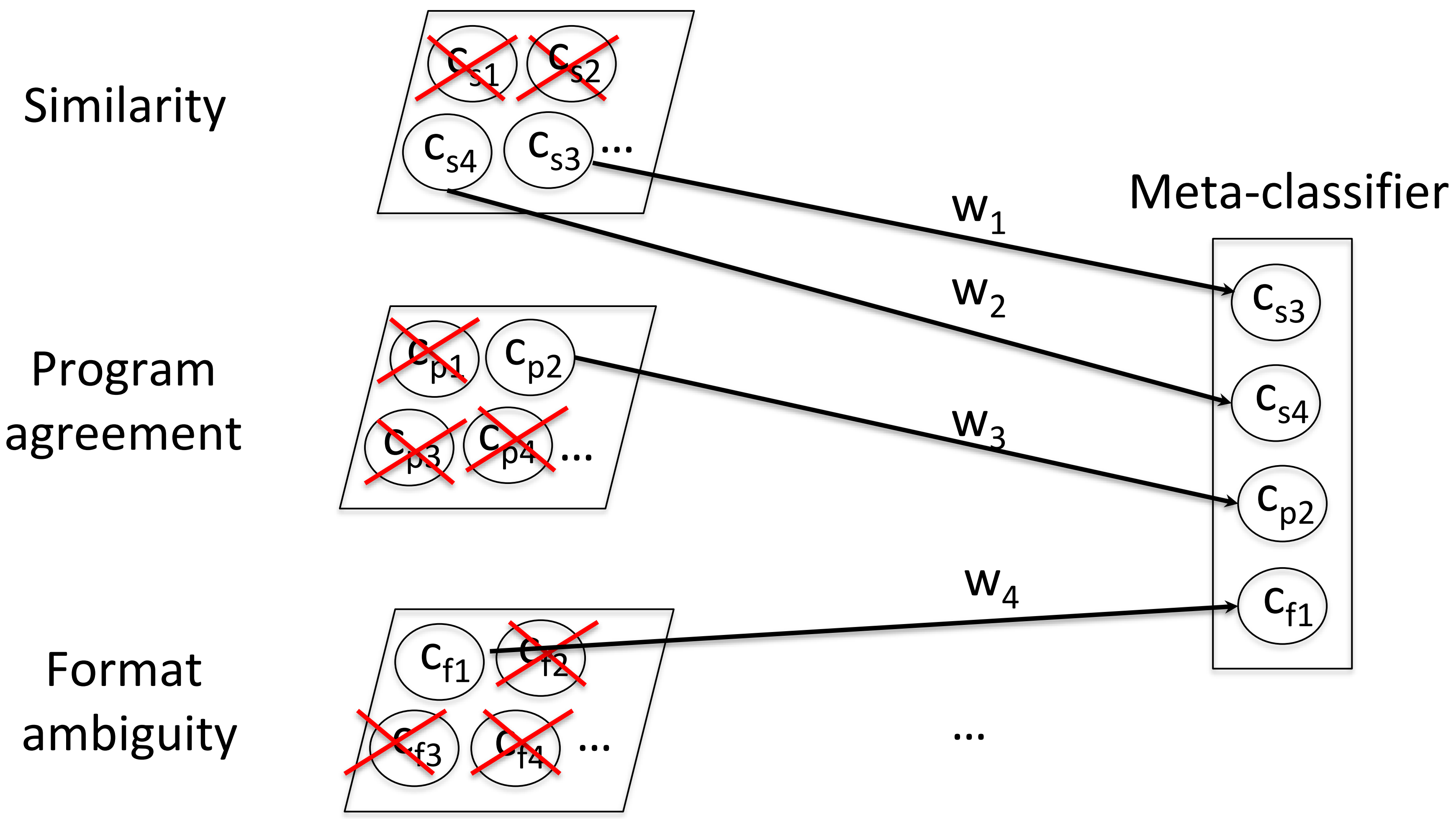
Program: (LWRD, ‘)’, 1)
Input: 2008 Mitsubishi Galant ES $7500 (Sylmar CA) pic

- Recommend potentially incorrect records
  - Learn a binary meta-classifier

Ex:

Raw	Transformed
11”H x 6”	11
30 x 46”	30 x 46
…	…

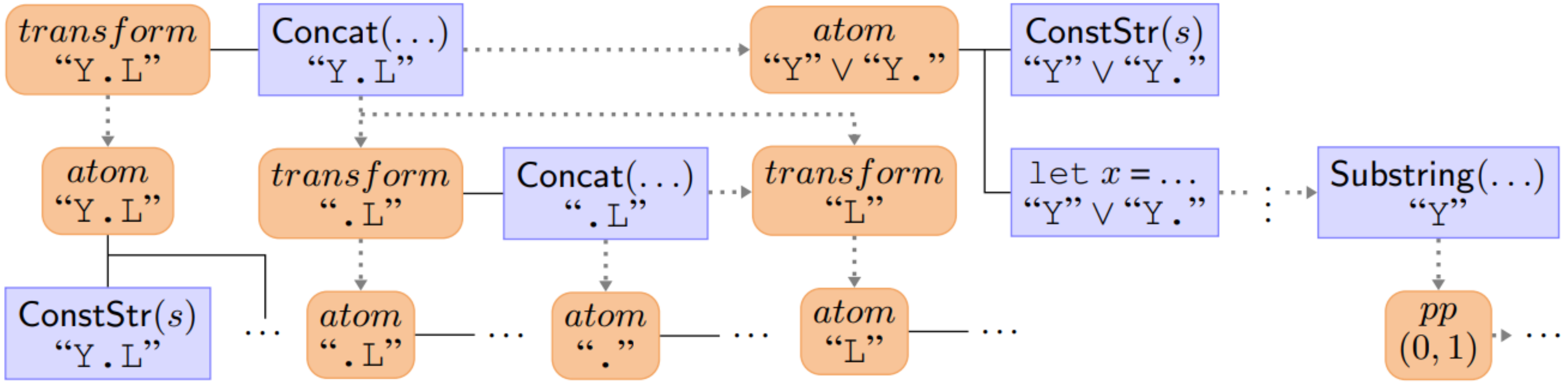
Learning the Meta-classifier



LEARNING TO PROGRAMMING-BY-EXAMPLE

Planning as a Searching Problem

- Planning Problem can be solved by searching




```
// Nonterminals
@start string transform := atom | Concat(atom, transform);
string atom := ConstStr(s)
               | let string x = std.Kth(inputs, k) in Substring(x, pp);
Tuple<int, int> pp := std.Pair(pos, pos) | RegexOccurrence(x, r, k);
int pos := AbsolutePosition(x, k) | RegexPosition(x, std.Pair(r, r), k);
// Terminals
@input string[] inputs;     string s;     int k;     Regex r;
```

Why not seq2seq ?

- We have input sequence: raw strings
- We have output sequence: user-provided strings
- Seems like a perfect seq2seq

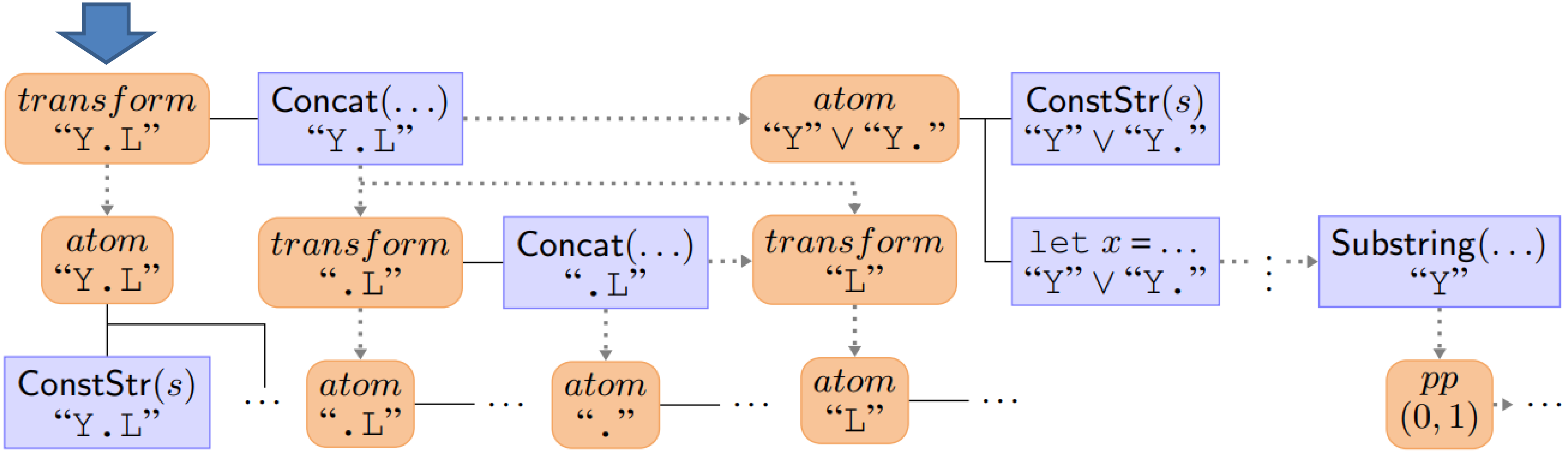
But it is not

Raw	Transformed
10“ H x 8” W	10
11”H x 6”	11
…	…
30 x 46”	30 x 46

Planning as a Searching Problem

- Planning Problem can be solved by searching

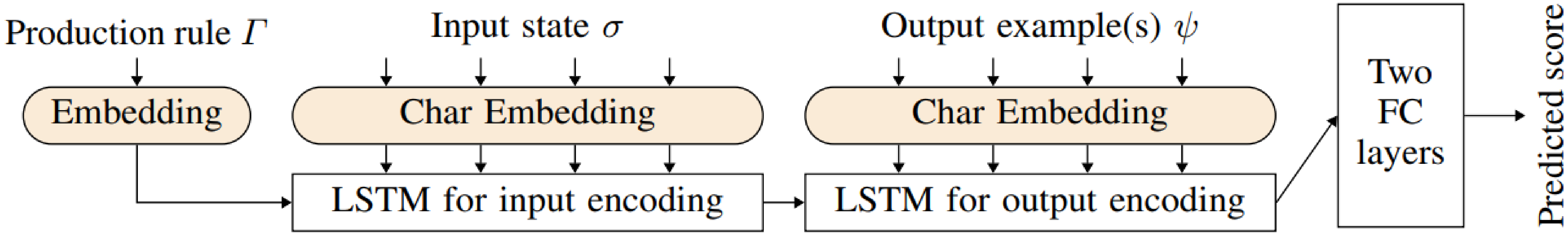
Predicting next state at a given state



Learning to Search

- Benefits:
  - Much easier to obtain training examples: synthesis training data
  - Search space can be much smaller than word space
  - Classification is easier to generalize

```
// Nonterminals
@start string transform := atom | Concat(atom, transform);
string atom := ConstStr(s)
             | let string x = std.Kth(inputs, k) in Substring(x, pp);
Tuple<int, int> pp := std.Pair(pos, pos) | RegexOccurrence(x, r, k);
int pos := AbsolutePosition(x, k) | RegexPosition(x, std.Pair(r, r), k);
// Terminals
@input string[] inputs;     string s;     int k;     Regex r;
```



John Morcos, Ziawasch Abedjan, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker

LEARNING TRANSFORMATIONS FROM WEB TABLES

Other Types of Transformations

How to fill in this table ?

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
HBE	?
IST	?
FRA	?
BOS	?
DFW	?

What if we have this ?

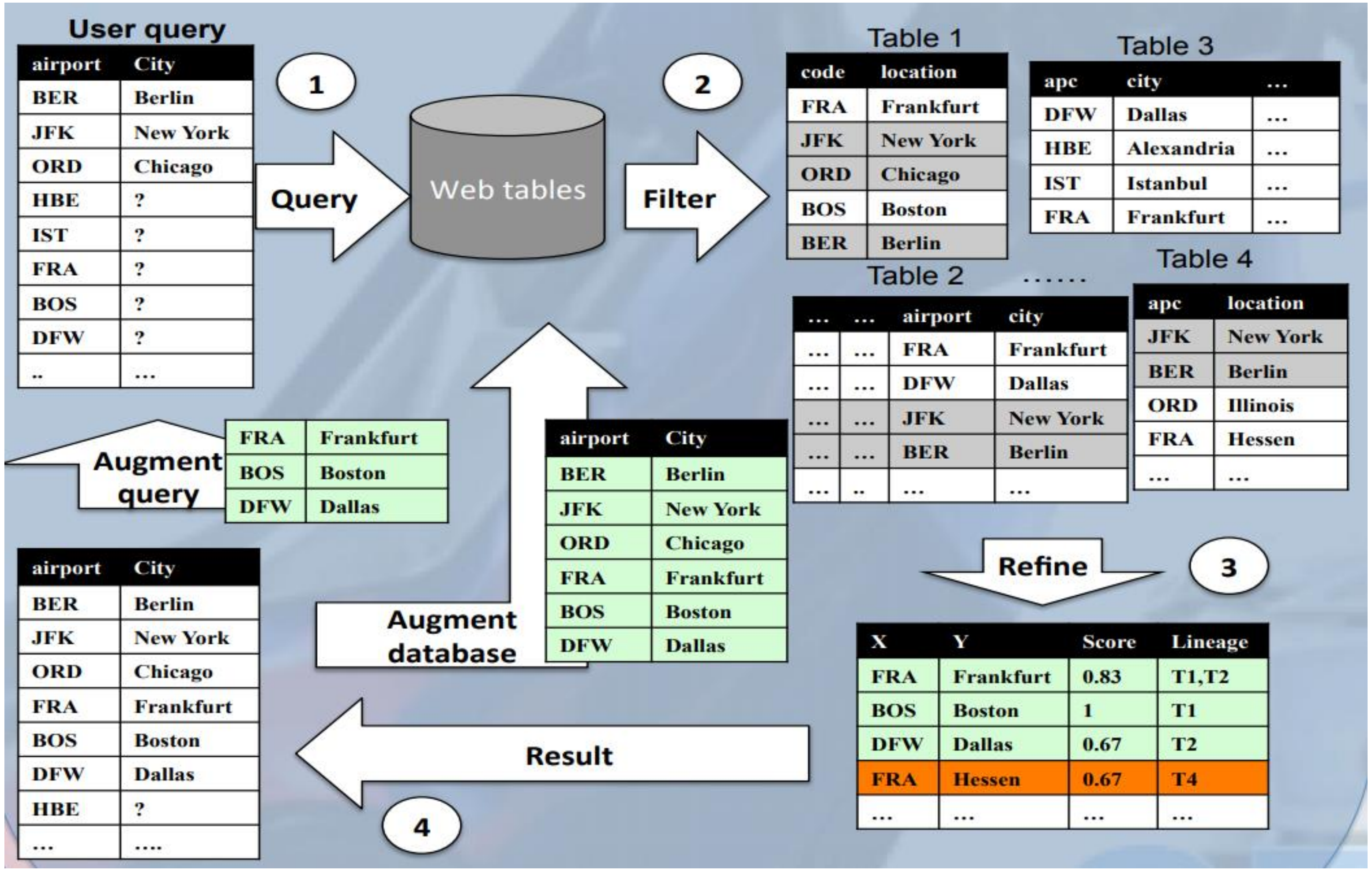
CDTFA-810-FTH (S2B) REV. 1 (10-17)

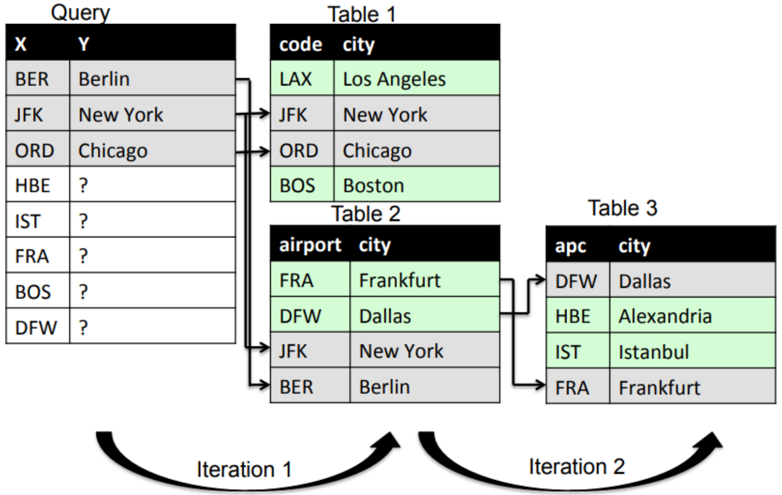
STATE OF CALIFORNIA
CALIFORNIA DEPARTMENT OF
TAX AND FEE ADMINISTRATION

Airport Code Table
(Sorted by Airport Code)

APC	City	Name
L84	Lost Hills	Lost Hills Airport
L88	New Cuyama	New Cuyama Airport
L90	Ocotillo Wells	Ocotillo Airport
L94	Tehachapi	Mountain Valley Airport
LAX	Los Angeles	Los Angeles International Airport
LGB	Long Beach	Long Beach Airport (Daugherty Field)
LHM	Lincoln	Lincoln Regional Airport (Karl Harder Field)
LLR	Little River	Little River Airport
LPC	Lompoc	Lompoc Airport
LSN	Los Banos	Los Banos Municipal Airport
LVK	Livermore	Livermore Municipal Airport
M45	Markleeville	Alpine County Airport
M90	Mendota	Mendota Airport
MAE	Madera	Madera Municipal Airport
MCC	Sacramento	McClellan Airfield (was McClellan AFB)
MCE	Merced	Merced Municipal Airport (MacReady Field)
MER	Atwater	Castle Airport
MHR	Sacramento	Sacramento Mather Airport
MHV	Mojave	Mojave Airport
MIT	Shafter	Shafter Airport (Minter Field)
MMH	Mammoth Lakes	Mammoth Yosemite Airport
MOD	Modesto	Modesto City-County Airport (Harry Sham Field)
MPI	Mariposa	Mariposa-Yosemite Airport

Extract Transformations from Web Tables





IMPUTING MISSING VALUES

Imputing Values to Missing Data

- In federated data, between 30%-70% of the data points will have at least one missing attribute - data wastage if we ignore all records with a missing value
- Remaining data is seriously biased
- Lack of confidence in results
- Understanding pattern of missing data unearths data integrity issues

Missing Value Imputation

- Standalone imputation
  - Mean, median, other point estimates
  - Assume: Distribution of the missing values is the same as the non-missing values.
  - Does not take into account inter-relationships
  - Introduces bias
  - Convenient, easy to implement

Missing Value Imputation

- Better imputation - use attribute relationships
- Assume : all prior attributes are populated

X1	X2	X3	X4	X5
1.0	20	3.5	4	.
1.1	18	4.0	2	.
1.9	22	2.2	.	.
0.9	15	.	.	.

- Common techniques
  – Regression (parametric)

Missing Value Imputation – Linear Regression

- Regression method
  - Use linear regression, sweep left-to-right

    X3=a+b*X2+c*X1;

    X4=d+e*X3+f*X2+g*X1, and so on
  - X3 in the second equation is estimated from the first equation if it is missing

DATA CLEANING TOOL - OPENREFINE

OpenRefine

- Powerful tool that can be effectively used for data cleansing
- Cleans and transforms raw data, linking it with web services and databases
- Very easy to use and has a web interface
- Freely available and works well with any browser