CS839:
Probabilistic Graphical Models

Lecture 23: Applications in Data Management

Theo Rekatsinas

Logistics

1. Project presentations next Tuesday

2. 10 Groups: 10 - 15 mins presentation per group (We will run late)

3. Things to cover:

- *What is the problem?*
- *Why is it interesting and important?*
- *Why is it hard? What are the baselines* (E.g., why do naive approaches fail?)
- *Why hasn't it been solved before?* (Or, what's wrong with previous proposed solutions? How does yours differ?)
- *What are the key components of your approach and results?*

Snorkel + Data Programming

MOTIVATION:

In practice, training data is often:

- *The* **bottleneck**
- *The* practical injection point for domain knowledge

KEY IDEA:

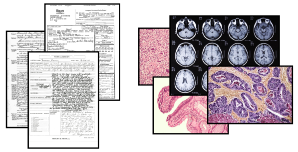
We can use ***higher-level, weaker*** supervision to ***program*** ML models

Outline

- **The Labeling Bottleneck:** *The new pain point of ML*
- **Data Programming + Snorkel:** *A framework for weaker, more efficient supervision*
- **In practice:** *Empirical results & user studies*

The ML Pipeline Pre-Deep Learning

Collection

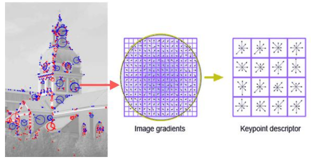


Labeling

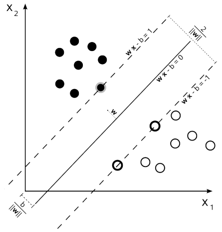
True

False

Feature Engineering

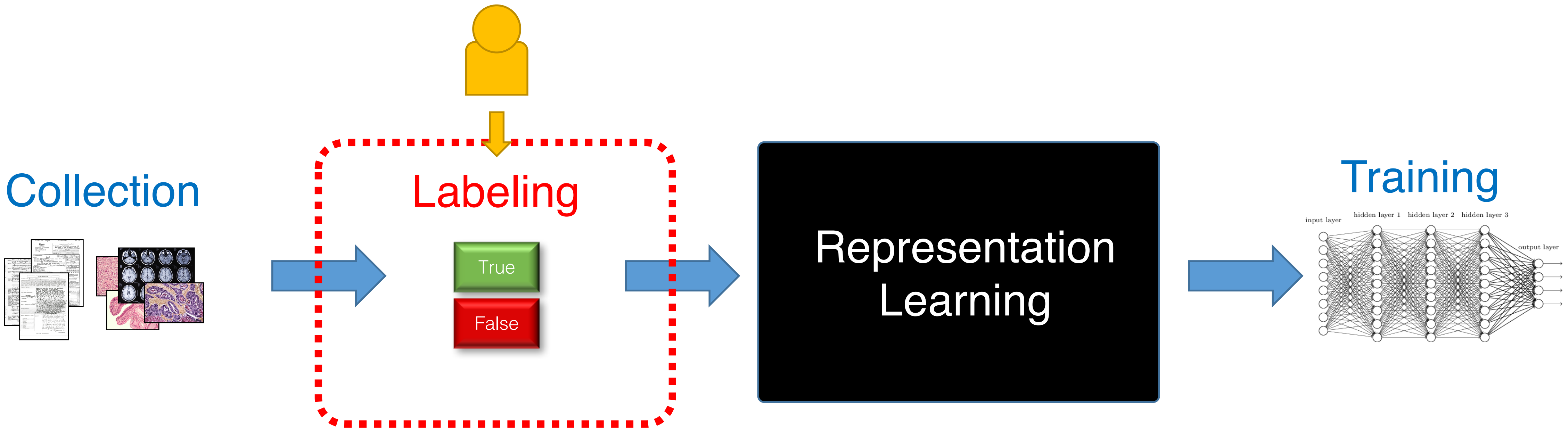


Training



Feature engineering *used to* be the bottleneck…

The ML Pipeline Today



New pain point, new injection point

Training Data: Challenges & Opportunities

- Expensive & Slow:
  - *Especially when domain expertise needed*

- Static:
  - *Real-world problems change; hand-labeled training data does not.*

- An opportunity to inject domain knowledge:
  - *Modern ML models are often too complex for hand-tuned structures, priors, etc.*

How do we get—*and use*—training data more effectively?

Data Programming + Snorkel

A Framework + System for Creating Training Data with Weak Supervision

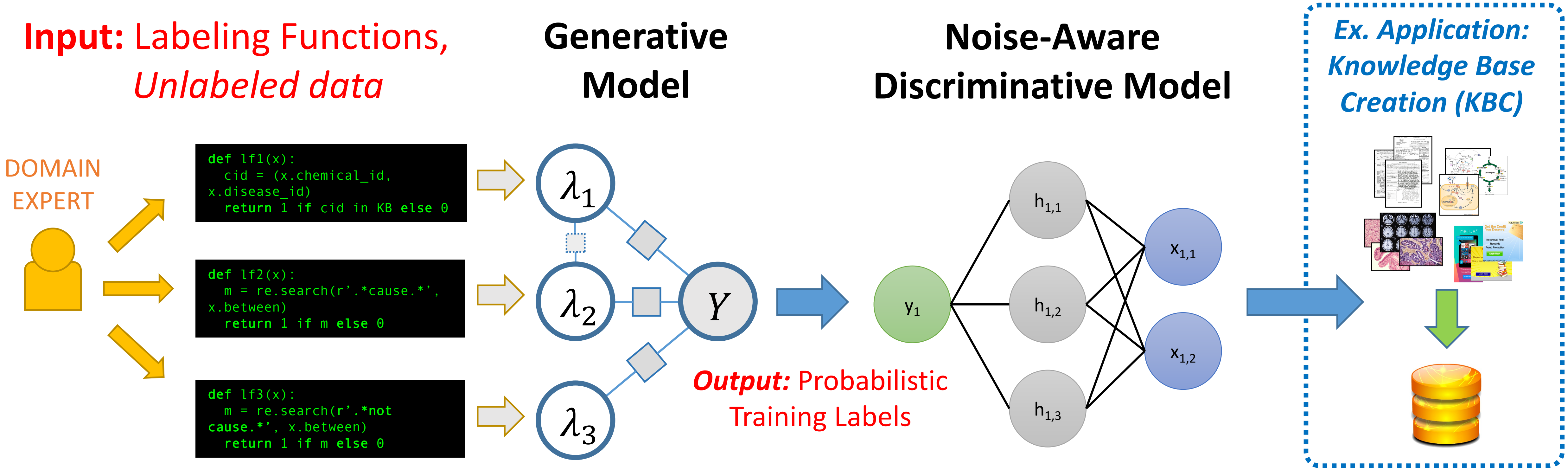
NIPS 2016 **SIGMOD (Demo) 2017**

KEY IDEA:

Get users to provide ***higher-level (but noisier)*** supervision,

Then model & de-noise it (using *unlabeled* data) to train **high-quality** models

Data Programming Pipeline in Snorkel



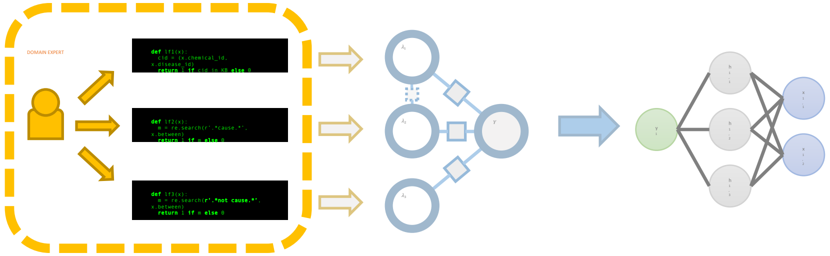
1 Users write *labeling functions* to generate noisy labels

2 We model the labeling functions' behavior to de-noise them

3 We use the resulting prob. labels to train a model

Surprising Point:

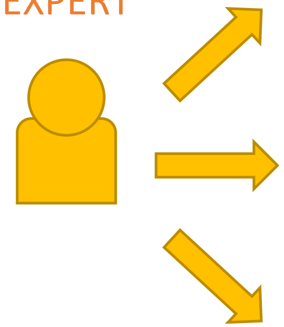
No hand-labeled training data!



Step 1: Writing Labeling Functions

A Unifying Framework for Expressing *Weak Supervision*

DOMAIN EXPERT



```
def lf1(x):
  cid = (x.chemical_id,
x.disease_id)
  return 1 if cid in KB else 0
```

```
def lf2(x):
  m = re.search(r’.*cause.*’,
x.between)
  return 1 if m else 0
```

```
def lf3(x):
  m = re.search(r’.*not
cause.*’, x.between)
  return 1 if m else 0
```

Example: Chemical-Disease Relation Extraction from Text

Helix Group

TITLE:
Myasthenia gravis presenting as weakness after magnesium administration.
ABSTRACT:
We studied a patient with no prior history of neuromuscular disease who became virtually quadriplegic after parenteral magnesium administration for preeclampsia. The serum magnesium concentration was 3.0 mEq/L, which is usually well tolerated. The magnesium was stopped and she recovered over a few days. While she was weak, 2-Hz repetitive stimulation revealed a decrement without significant facilitation at rapid rates or after exercise, suggesting postsynaptic neuromuscular blockade. After her strength returned, repetitive stimulation was normal, but single fiber EMG revealed increased jitter and blocking. Her acetylcholine receptor antibody level was markedly elevated. Although paralysis after magnesium administration has been described in patients with known myasthenia gravis, it has not previously been reported to be the initial or only manifestation of the disease. Patients who are unusually sensitive to the neuromuscular effects of magnesium should be suspected of having an underlying disorder of neuromuscular transmission.

- We define candidate entity mentions:
  - **Chemicals**
  - **Diseases**
- Goal: Populate a relational schema with *relation mentions*

ID	Chemical	Disease	Prob.
00	magnesium	Myasthenia gravis	0.84
01	magnesium	quadriplegic	0.73
02	magnesium	paralysis	0.96

KNOWLEDGE BASE (KB)

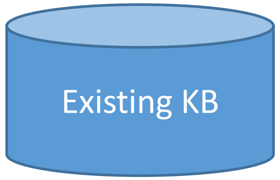
Labeling Functions

- Traditional “distant supervision” rule relying on external KB

```
def lf1(x):
  cid =(x.chemical_id,x.disease_id)
  return 1 if cid in KB else 0
```

”Chemical A is found to cause disease B under certain conditions…”

`Label = TRUE`



`Contains (A,B)`

This is likely to be true… *but*

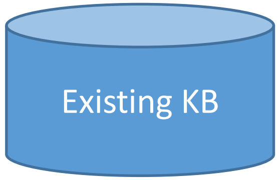
Labeling Functions

- Traditional "distant supervision" rule relying on external KB

```
def lf1(x):
  cid =(x.chemical_id,x.disease_id)
  return 1 if cid in KB else 0
```

"Chemical A was found on the floor near a person with disease B..."

`Label = TRUE`



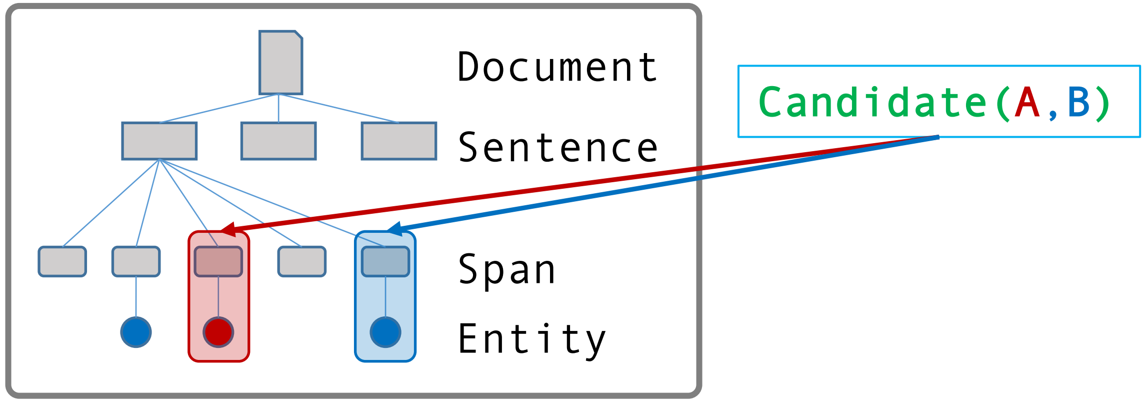
`Contains (A,B)`

...can be false!

We will learn the accuracy of each LF (next)

Writing Labeling Functions in Snorkel

- Labeling functions take in `Candidate` objects:



CONTEXT HIERARCHY

Key Point: *Supervision as code*

- Three levels of abstraction for writing LFs in Snorkel:
  - Python code

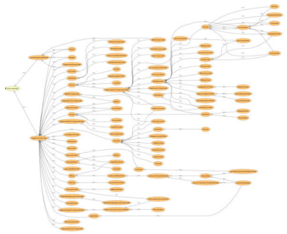
```
def lf1(x):
  cid =(x.chemical_id,x.disease_id)
  return 1 if cid in KB else 0
```

  - LF templates

```
lf1 = LF_DS(KB)
```

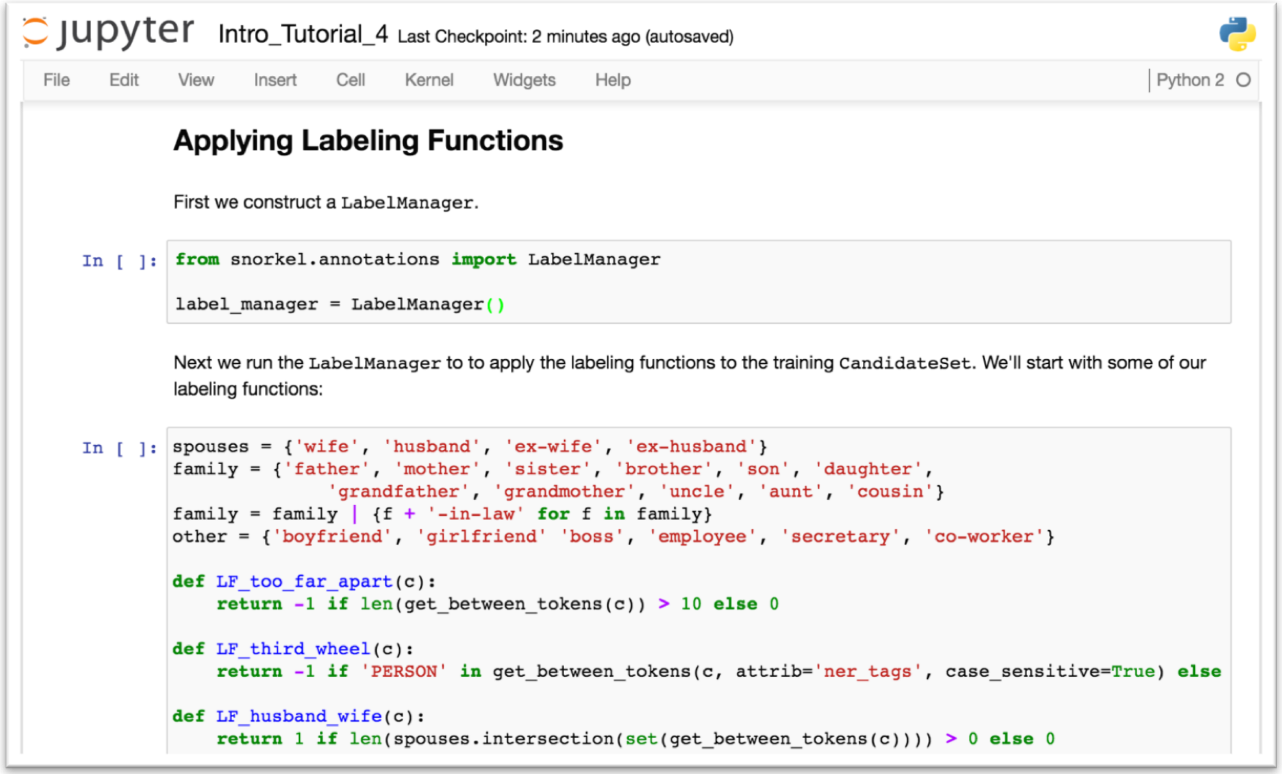
  - LF generators

```
for lf in LF_DS_hier(KB, cut_level=2):
    yield lf
```



A knowledge base (KB) with hierarchy

Supported by Simple Jupyter Interface



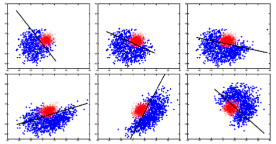
snorkel.stanford.edu

Broader Perspective:

A Template for Weak Supervision

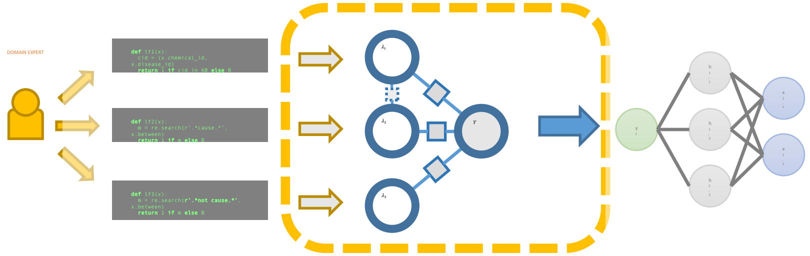
A Unifying Method for Weak Supervision

- Distant supervision
- Crowdsourcing
- Weak classifiers
- Domain heuristics / rules

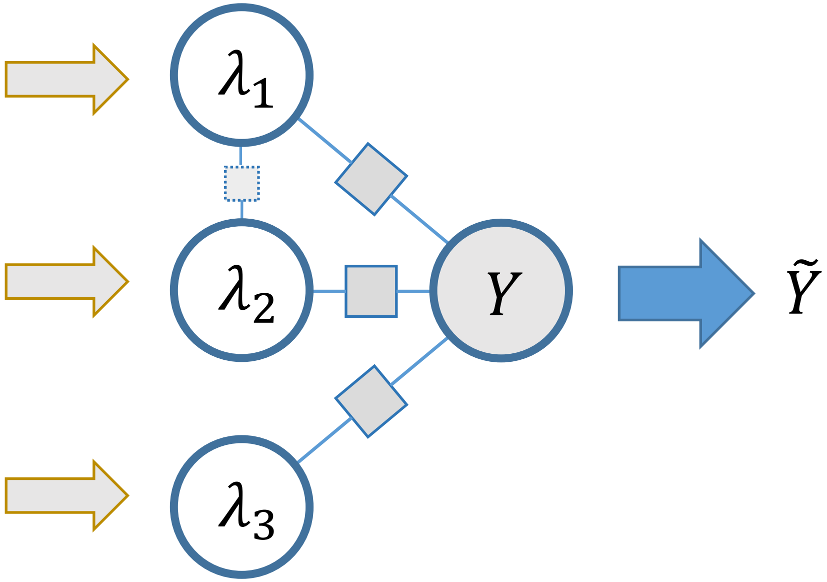


$$\lambda : X \mapsto Y \cup \{\emptyset\}$$

How to handle such a diversity of weak supervision sources?



Step 2: Modeling Weak Supervision

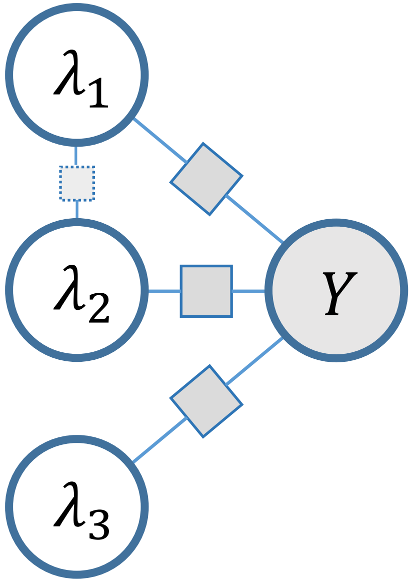


Weak Supervision: Core Challenges

- Unified input format

- Modeling
    - Accuracies of sources
    - Correlations between sources
    - Expertise of sources

- Using to train a wide range of models

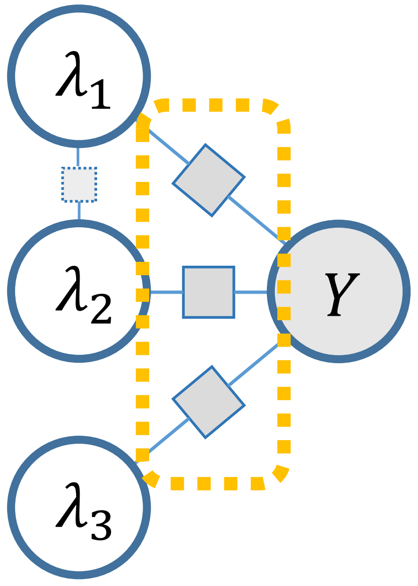


Weak Supervision: Core Challenges

- **Unified input format**

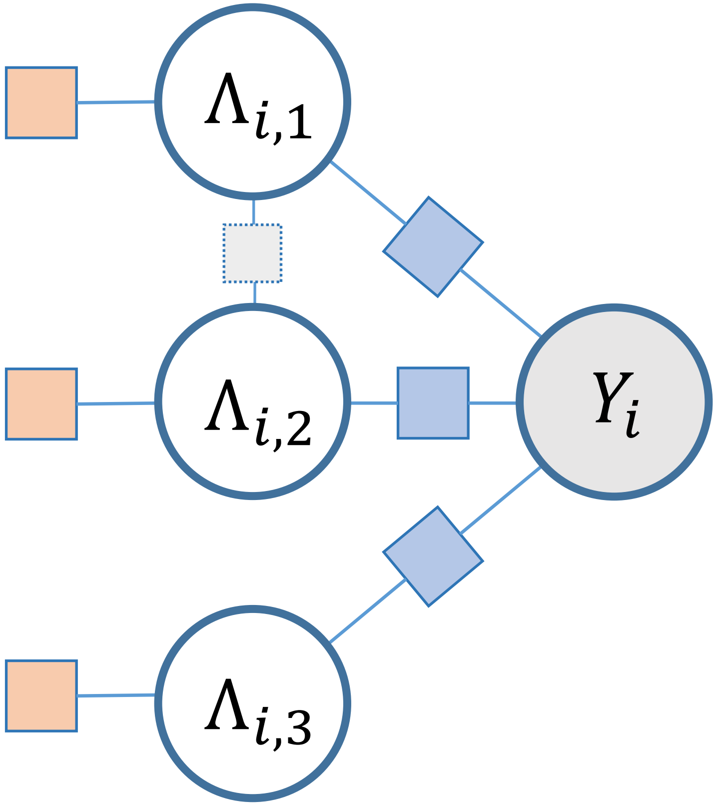
- Modeling
  - **Accuracies of sources** **NIPS 2016**
  - Correlations between sources
  - Expertise of sources

- **Using to train a wide range of models**



Intuition: We use agreements / disagreements to learn without ground truth

Basic Generative Labeling Model



Labeling propensity:

$\beta_j = p_\theta(\Lambda_{i,j} \neq \emptyset)$ $\quad f_j^{lab}(\Lambda_i, Y_i) = \exp(\theta_j^{lab} \Lambda_{i,j}^2)$

Accuracy:

$\alpha_j = p_\theta\left(\Lambda_{i,j} = Y_i \mid Y_i, \Lambda_{i,j} \neq \emptyset\right)$

$f_j^{acc}(\Lambda_i, Y_i) = \exp(\theta_j^{acc} \Lambda_{i,j} Y_i)$

Correlations ICML 2017

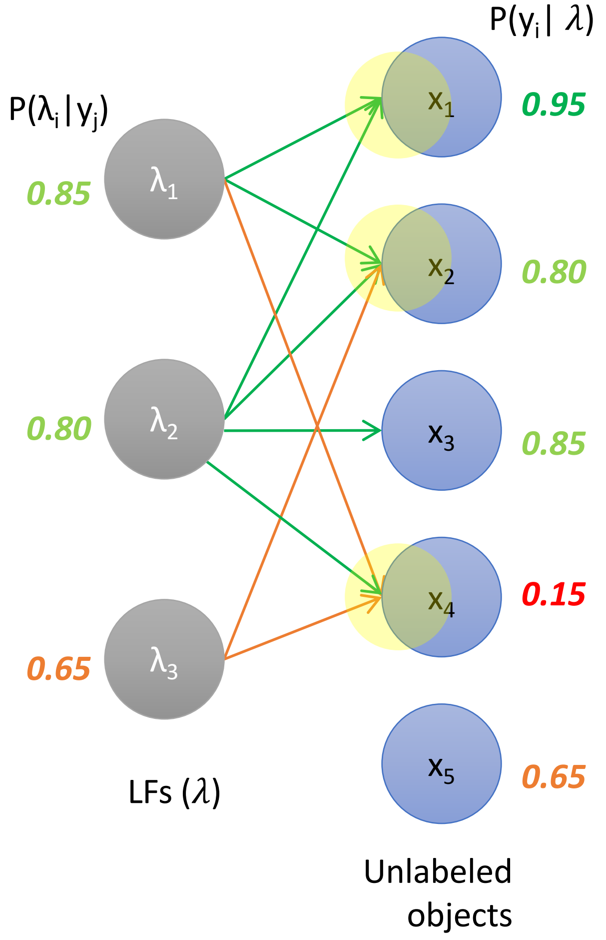
Intuition: Learning from Disagreements

Learn the model $\pi = P(y, \Lambda)$ using MLE

- LFs have a hidden ***accuracy parameter***
- Intuition: Majority vote--estimate labeling function accuracy based on overlaps / conflicts
    - Similar to ***crowdsourcing but different scaling.***
    - *small number of LFs, large number of labels each*

Produce a set of *noisy* training labels

$$\mu_\pi(y, \lambda) = P_{(y,\Lambda)\sim\pi}(y \mid \Lambda = \lambda(x))$$



Step 2: Training a Noise-Aware Model

In a supervised learning setting, we would learn from ground-truth labels:

$$\hat{w} = \text{argmin}_w \frac{1}{N} \sum_{i=1}^{N} l(w, x^{(i)}, y^{(i)}) \qquad T = \{(x_1, 0), (x_2, 1), (x_3, 0), \dots\}$$

Here, we learn from the *noisy* labels:

$$\hat{w} = \text{argmin}_w \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{(\boldsymbol{y},\boldsymbol{\Lambda})\sim\boldsymbol{\pi}}[l(w, x^{(i)}, y^{(i)} = y)] \qquad T = \{(x_1, 0.1), (x_2, 0.6), (x_3, 0.3), \dots\}$$

Only requires simple tweak to loss function works over ***many models*** including Logistic Regression, SVMs and LSTMs.

Theory: Scaling with *Unlabeled* Data

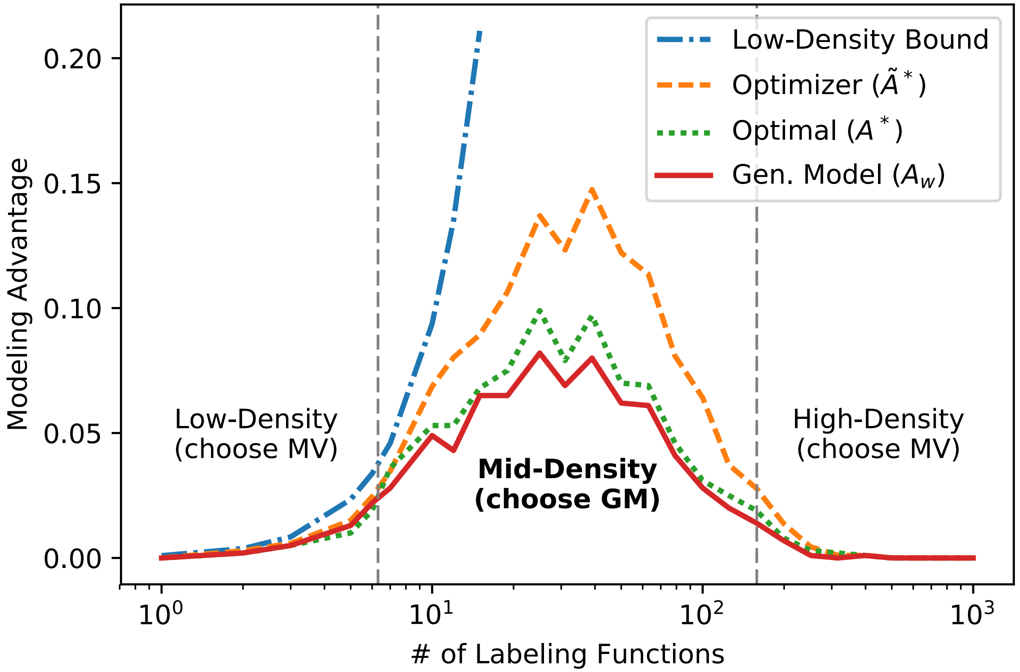
- We show that with:
  - $O(1)$ labeling functions of sufficient quality / expressiveness
  - $\tilde{O}(\epsilon^{-2})$ ***unlabeled*** training data points
  - → We get $O(\epsilon)$ generalization risk

This is the same asymptotic scaling as in supervised methods!

When is modeling the noise worthwhile?

- Can look at *label density:*
  - Low: Too sparse to beat MV
  - High: MV approaches optimal
  - Medium: Just right!

- Can use conditional decision rule to safely skip gen. modeling stage
  - E.g. during early LF dev cycles



Putting it All Back Together

Input: Labeling Functions, *Unlabeled data*

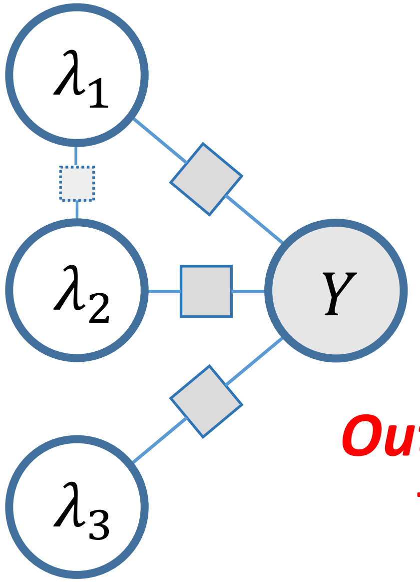
DOMAIN EXPERT

```
def lf1(x):
  cid = (x.chemical_id,
x.disease_id)
  return 1 if cid in KB else 0
```

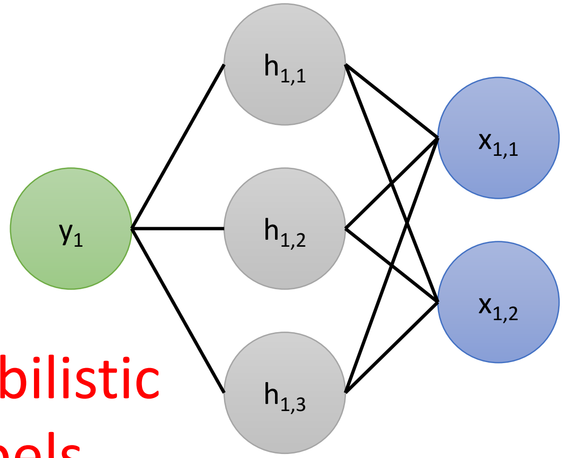
```
def lf2(x):
  m = re.search(r'.*cause.*',
x.between)
  return 1 if m else 0
```

```
def lf3(x):
  m = re.search(r'.*not
cause.*', x.between)
  return 1 if m else 0
```

Generative Model



Noise-Aware Discriminative Model



Output: Probabilistic Training Labels

1 Users write *labeling functions* to generate noisy labels

2 We model the labeling functions' behavior to de-noise them

3 We use the resulting prob. labels to train a model

How well does this work in practice?

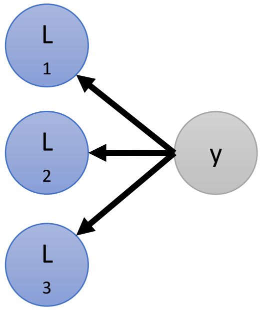
Empirical Results

Results on Chemical-Disease Relations

Precision: 25.5
Recall: 34.8
F1: **29.4**

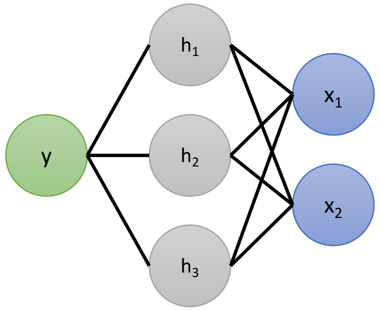
Distant Supervision

Precision: 52.3
Recall: 30.4
F1: **38.5**
+ 9.1



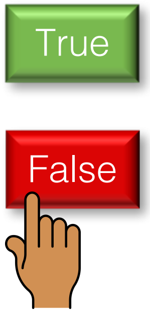
Generative Model

Precision: 38.8
Recall: 54.3
F1: **45.3**
+ 6.8



Discriminative Model

Precision: 39.9
Recall: 58.1
F1: **47.3**
+ 2.0



Hand Supervision

How easy is this to use in practice?

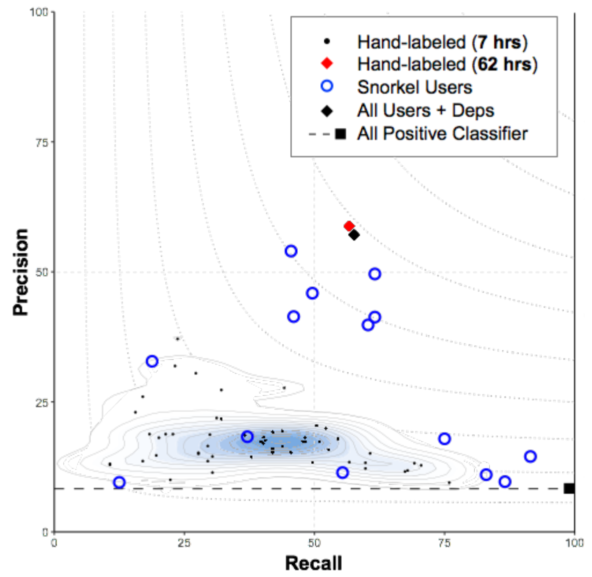
User Study

Snorkel User Study

We recently ran a Snorkel biomedical workshop in collaboration with the NIH Mobilize Center

15 companies and research groups attended

How well did these new Snorkel users do?



71% New Snorkel users matched or beat 7 hours of hand-labeling

2.8x Faster than hand-labeling data

45.5% Average improvement in model performance

Marta Gaia Zanchi
@medinnovo
Following

For a newbie, I write pretty darn good #Snorkel #MachineLearning labeling functions. Thanks @MobilizeCenter @jasonafries @stevebach :)

3rd Place Score

No machine learning experience
Beginner-level Python

Conclusion

- Snorkel provides a unifying framework for **combining and modeling *weak supervision***
  - Allows us to rapidly generate training data for modern ML models
  - Labeling functions: ***supervision as code***
- For more check out snorkel.stanford.edu: Code, tutorials, blogs, papers

snorkel.stanford.edu

Fonduer: Knowledge Base Construction from Richly Formatted Data

FONDUER

Knowledge bases are everywhere…

Unstructured Information

Knowledge Base Construction →

Structured Knowledge Base

KNOWLEDGE VAULT

yago select knowledge

ProbKB

IBM Watson

Product Graph

TextRunner/ ReVerb

NELL

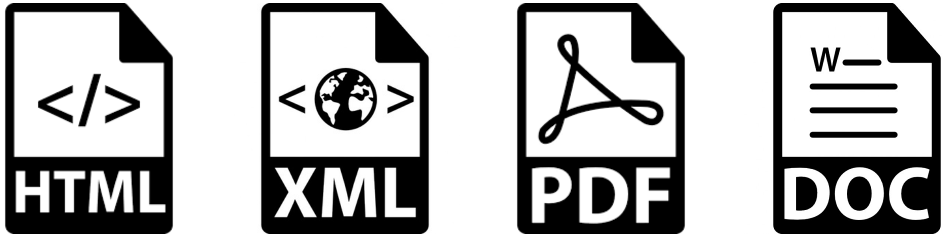
DeepDive

And many more…

But, troves of "richly formatted" information remains untapped

Richly formatted data

Richly formatted data: information is expressed via textual, structural, tabular, and visual cues.



Transistor Datasheet (PDF)

SMBT3904...MMBT3904

NPN Silicon Switching Transistors

- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation	P_{tot}		mV
$T_S \leq 71°C$		330	
$T_S \leq 115°C$		250	
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

Knowledge base construction from richly formatted data

Goal: extract maximum collector current from transistor datasheets

Transistor Datasheet

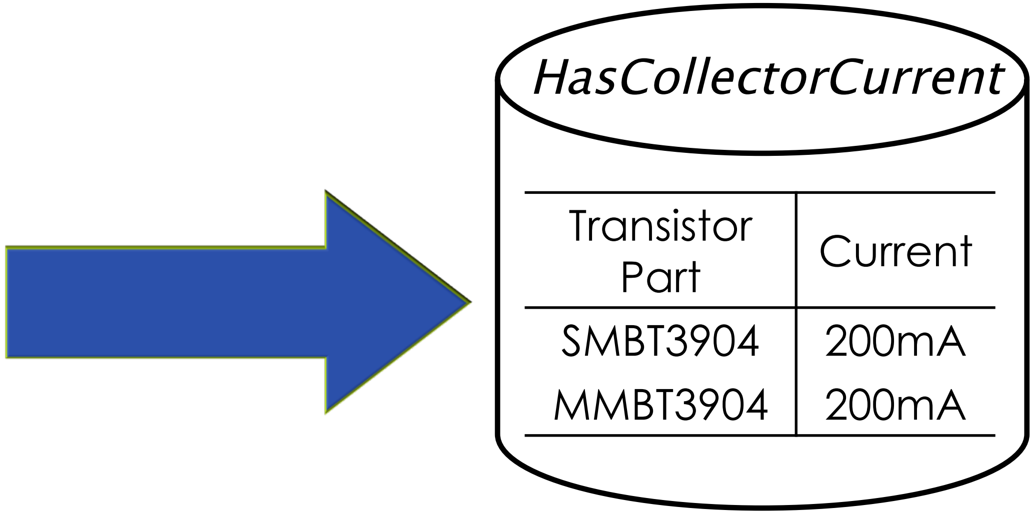
SMBT3904...MMBT3904

NPN Silicon Switching Transistors

- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

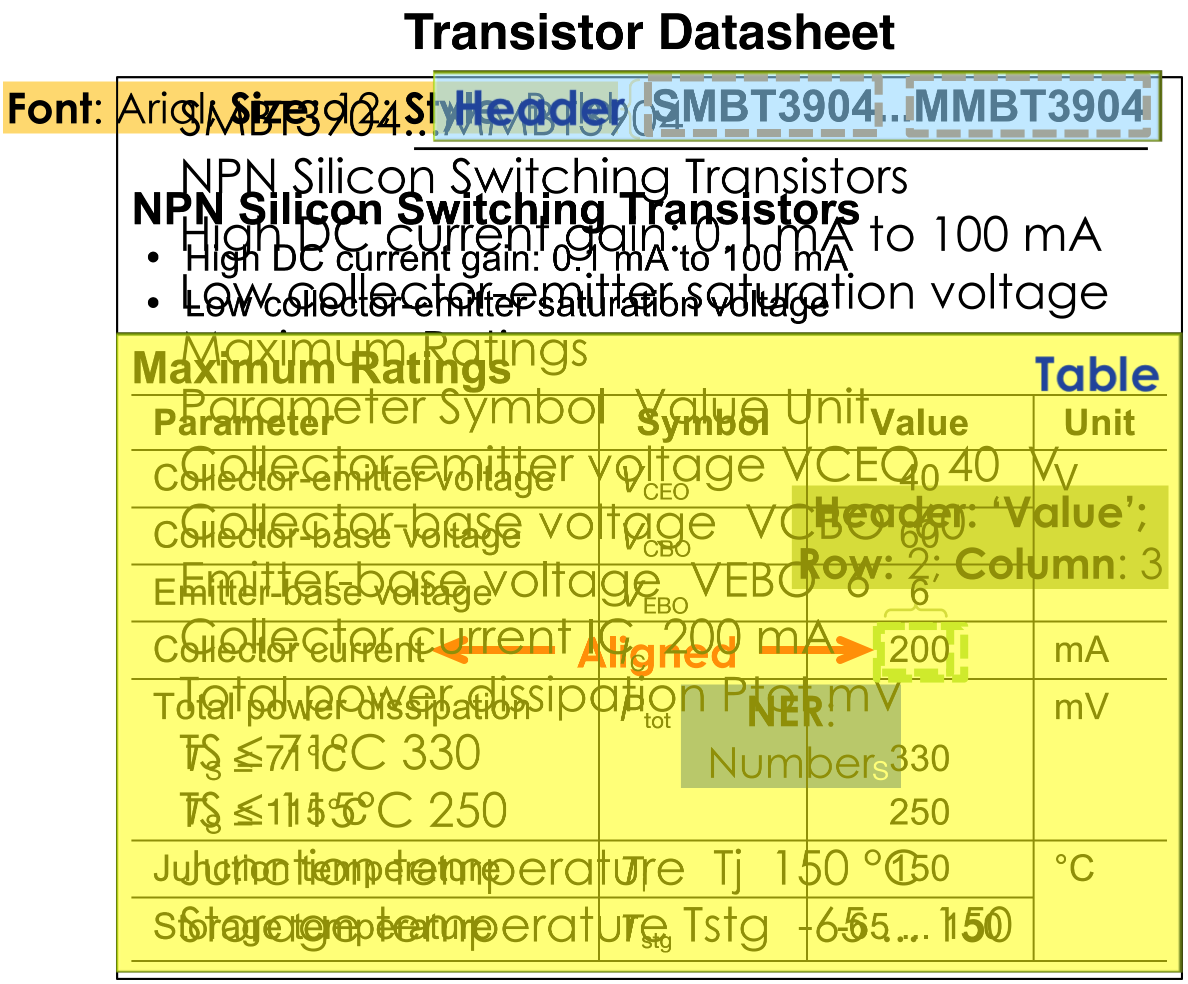
Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation	P_{tot}		mV
$T_S \leq 71°C$		330	
$T_S \leq 115°C$		250	
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	



Knowledge Base

Knowledge base construction from richly formatted data



In richly formatted data, semantics are expressed in textual, structural, tabular, and visual modalities throughout a document

Conventional approach 1: Filter out other modalities besides unstructured text

Conventional approach 2: Limit the context scope to sentences or tables.

Problem: Misses important relations if you neglect multimodal information

Up to 97% missed relations!

Deep learning is very successful in many domains

Andrej Karpathy Follow
Director of AI at Tesla. Previously Research Scientist at OpenAI and PhD student at Stanford. I like to train deep neural nets on large datasets.
Nov 11, 2017 · 8 min read

Software 2.0

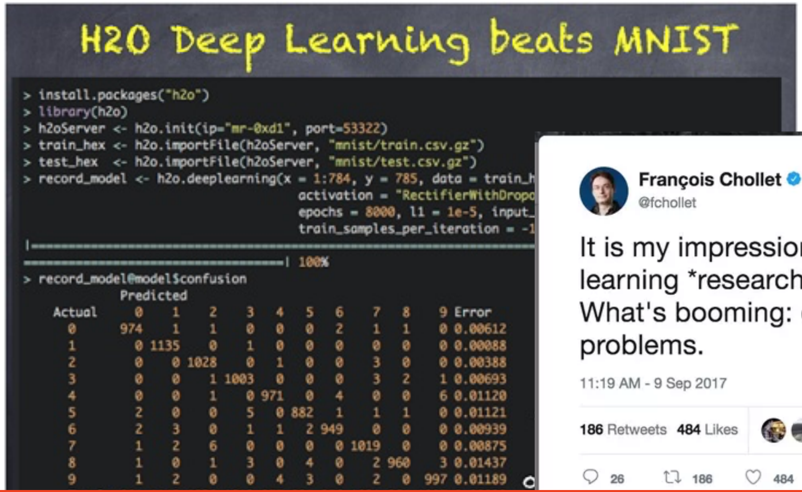
I sometimes se… tool in your
machine learn… work here or
there, and som… ons.
Unfortunately,
Neural networ
of a fundamen

Alibaba's artificial intelligence bot beats humans at reading in a first for machines

A deep neural network model developed by Alibaba has scored higher than humans in a reading comprehension test, paving the way for bots to replace people in customer service jobs

PUBLISHED : Monday, 15 January, 2018, 11:33am
UPDATED : Monday, 15 January, 2018, 12:17pm

SQuAD
The Stanford Question Answering Dataset



François Chollet
@fchollet
Following

It is my impression that the world of deep learning *research* is starting to plateau. What's booming: deploying DL to real-world problems.

11:19 AM - 9 Sep 2017

186 Retweets 484 Likes

26 186 484

KEY MOMENTS IN DEEP-LEARNING HISTORY 2014-2016

2014
JANUARY
Google acquires DeepMind, a startup specializing in combining deep learning and reinforcement learning, for $600 million.

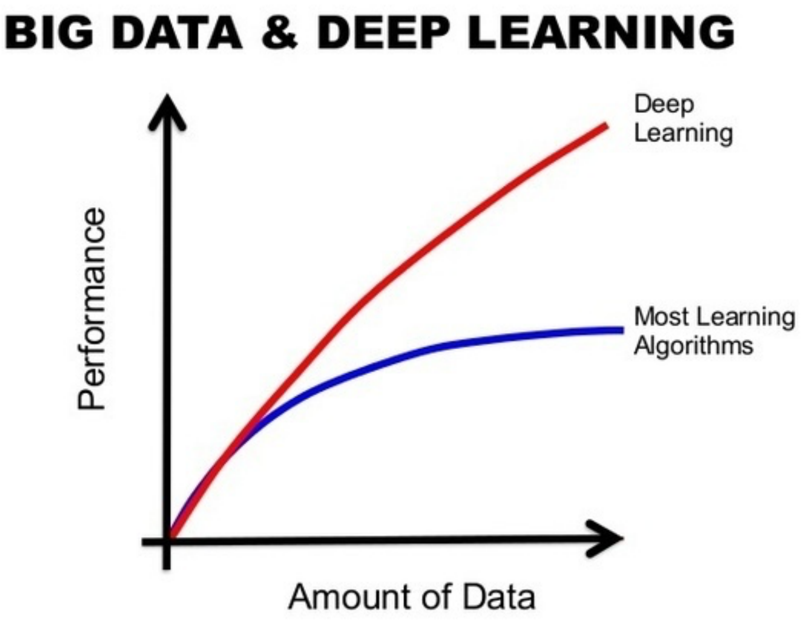
2015
DECEMBER
A team from Microsoft, using neural nets, outperforms a human on the ImageNet challenge.

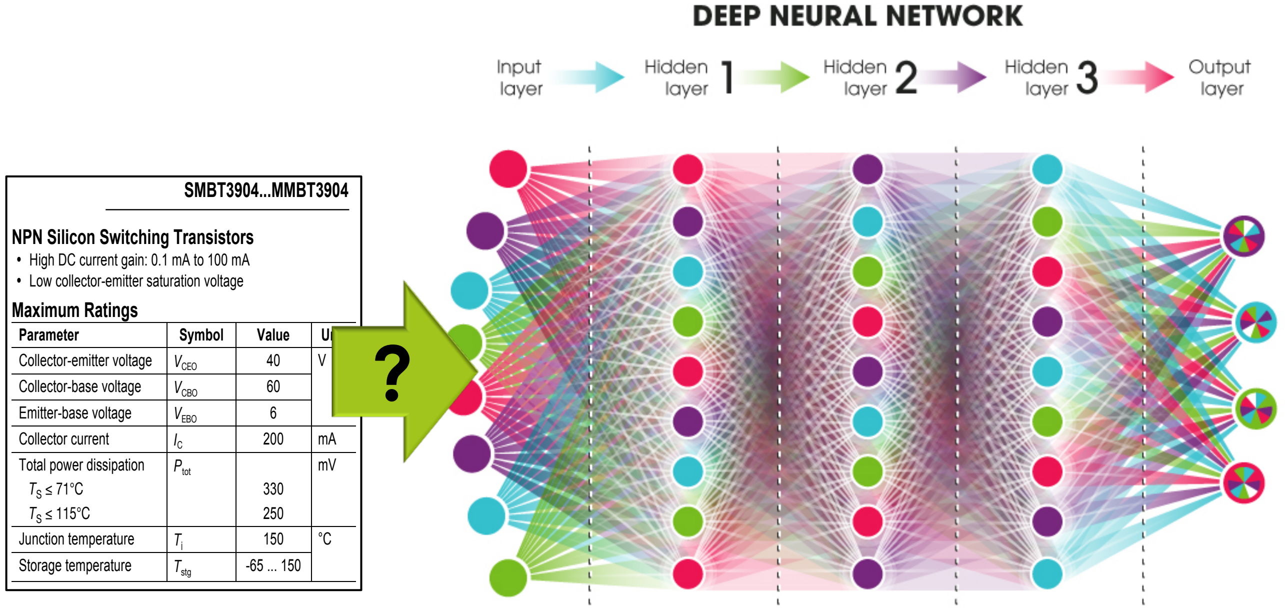
2016
MARCH
DeepMind's AlphaGo, using deep learning, defeats world champion **Lee Sedol** in the Chinese game of go, four games to one.

LEE JIN-MAN—AP PHOTO

Can we take advantage of this powerful tool and apply it to our problem?

Keys to utilizing deep learning





SMBT3904...MMBT3904

NPN Silicon Switching Transistors
- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Maximum Ratings

Parameter	Symbol	Value	U
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation	P_{tot}		mV
$T_S \leq 71°C$		330	
$T_S \leq 115°C$		250	
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

neuralnetworksanddeeplearning.com - Michael Nielsen, Yoshua Bengio, Ian Goodfellow, and Aaron Courville, 2016.

How do we gather enough labeled, richly formatted data?

How do we model the characteristics of richly formatted data in DL?

Fonduer

A *weakly supervised* deep learning framework for knowledge base construction from richly formatted data

Fonduer in practice!

FONDUER

DARPA MEMEX

accenture

Alibaba Group

Stanford MEDICINE

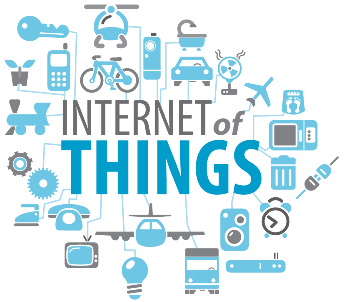
SING Stanford Information Networks Group

Macrostrat Lab

Anti-Human Trafficking

Search Engine

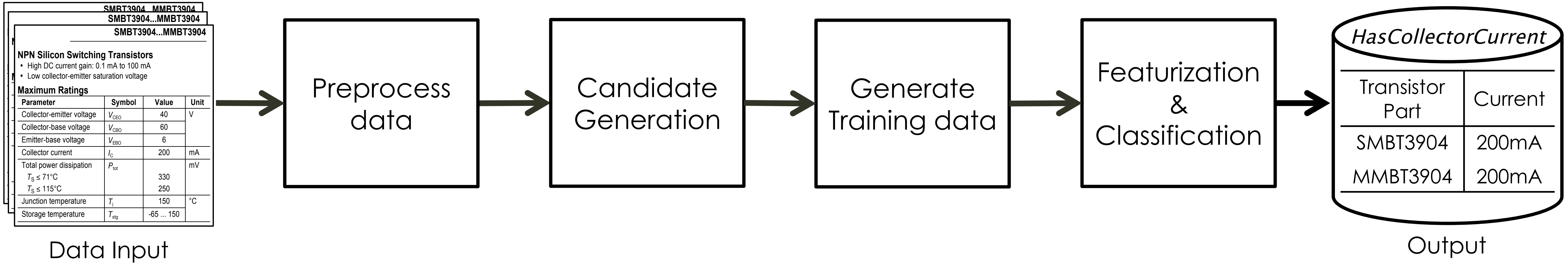
Genome-wide Association Studies



Internet of Things

Paleontology

Fonduer pipeline

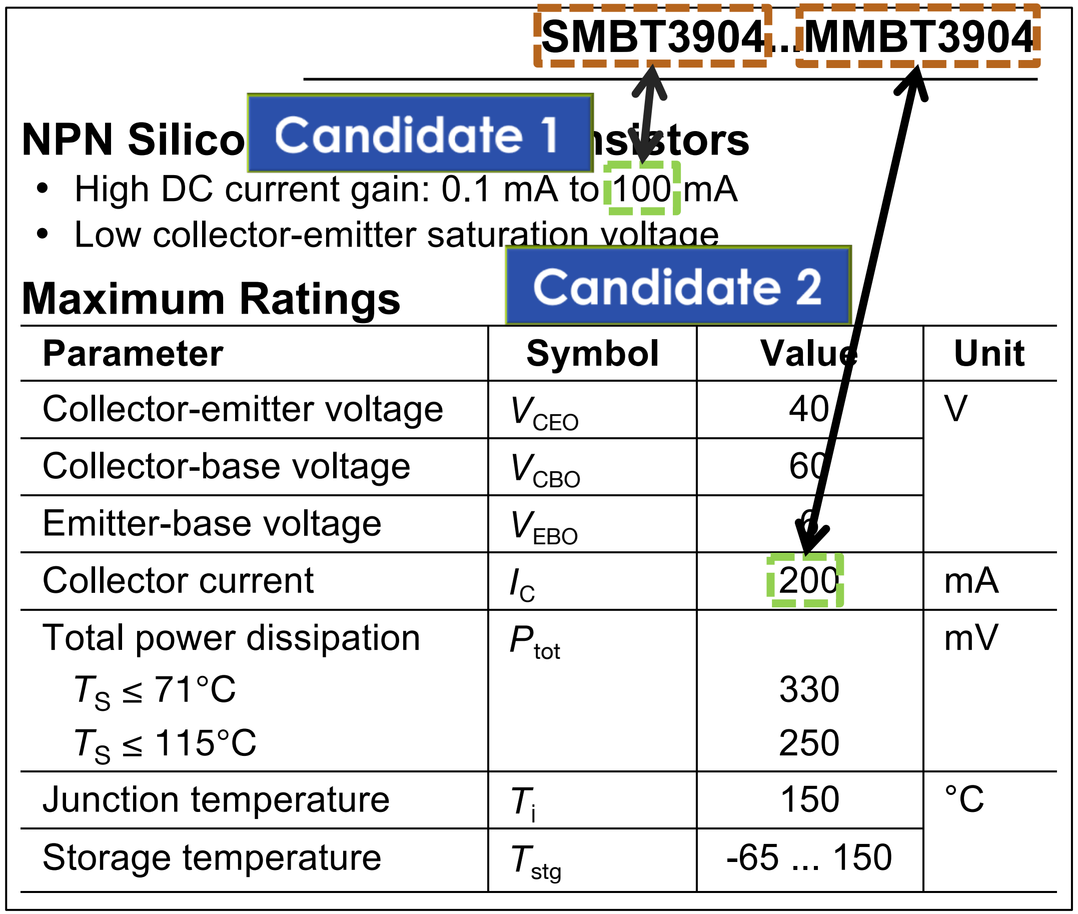


Generating richly formatted training data

Multimodal weak supervision

FONDUER

Transistor Datasheet



SMBT3904...MMBT3904

Candidate 1

NPN Silico nsistors

- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Candidate 2

Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation $T_S \leq 71°C$ $T_S \leq 115°C$	P_{tot}	 330 250	mV
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

Doc. level Candidates		Supervision: Manual	Supervision: Labeling function
SMBT3904	100	✗	✗
MMBT3904	200	✓	✓

Weak supervision: express any supervision signal via labeling functions to generate training data

```
# Check if current is in the same row with keyword `collector`
def in_the_same_row_with(candidate):
  if 'collector' in row_ngrams(candidate.current):
    return 1
  else: return -1
```

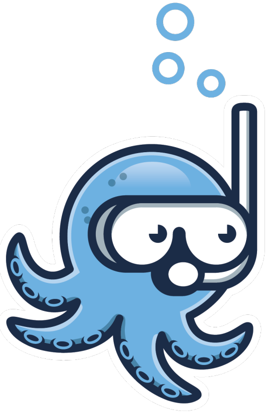
Modeling Weak Supervision

FONDUER

Doc. level Candidates		Multimodal Supervision: Vertically aligned with 'Value'	Multimodal Supervision: Row ngrams contain 'mA'	Multimodal Supervision: 'current' in sentence
SMBT3904	100	✗	∅	✓
SMBT3904	200	✓	✓	✗
SMBT3904	150	✓	✗	✗

∅ =Abstain

Intuition: Use agreements / disagreements to learn the accuracy of LFs without ground truth



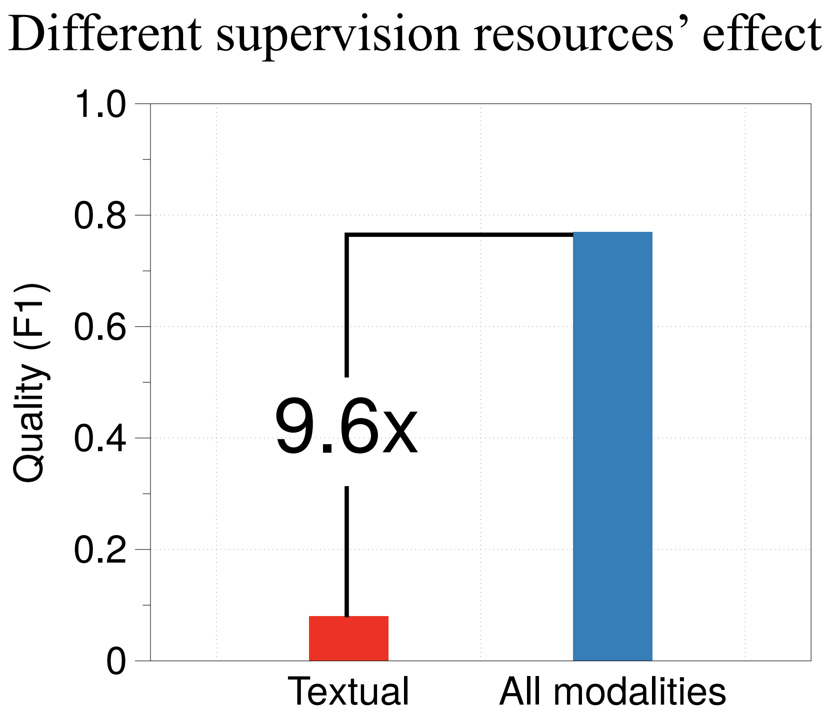
Data programming/MeTal

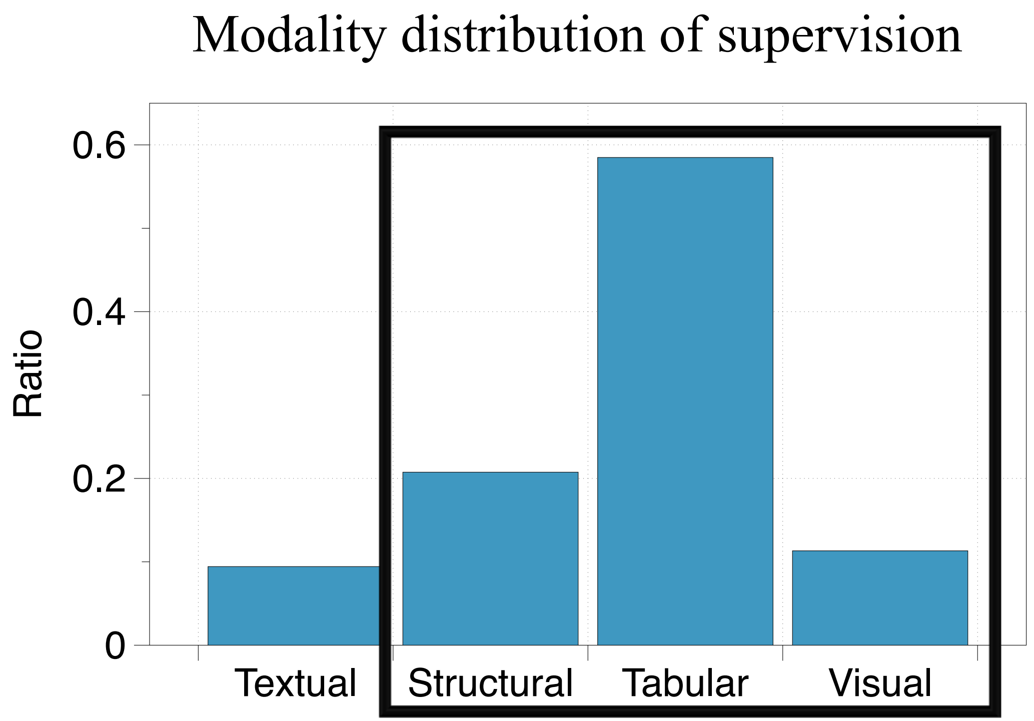
Output: Probabilistic Training Labels

SMBT3094	100	0.5
SMBT3094	200	0.85
SMBT3094	150	0.15

Multimodal supervision is key to quality

For transistor datasheets…





Users intuitively rely on multimodal information for supervision

Featurization and Classification for Richly Formatted Data

LSTM for Textual Information

Transistor Datasheet

SMBT3904...MMBT3904

NPN Silicon Switching Transistors

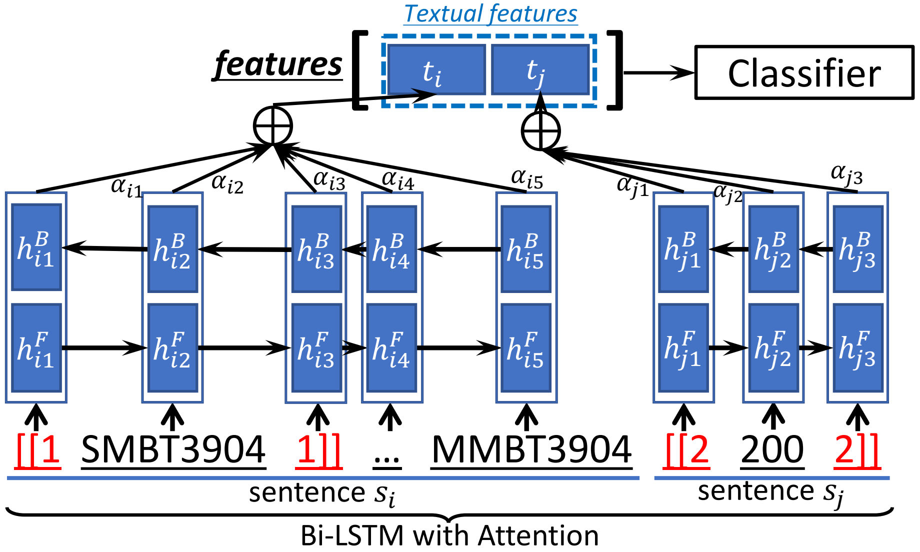
- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation	P_{tot}		mV
$T_S \leq 71°C$		330	
$T_S \leq 115°C$		250	
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

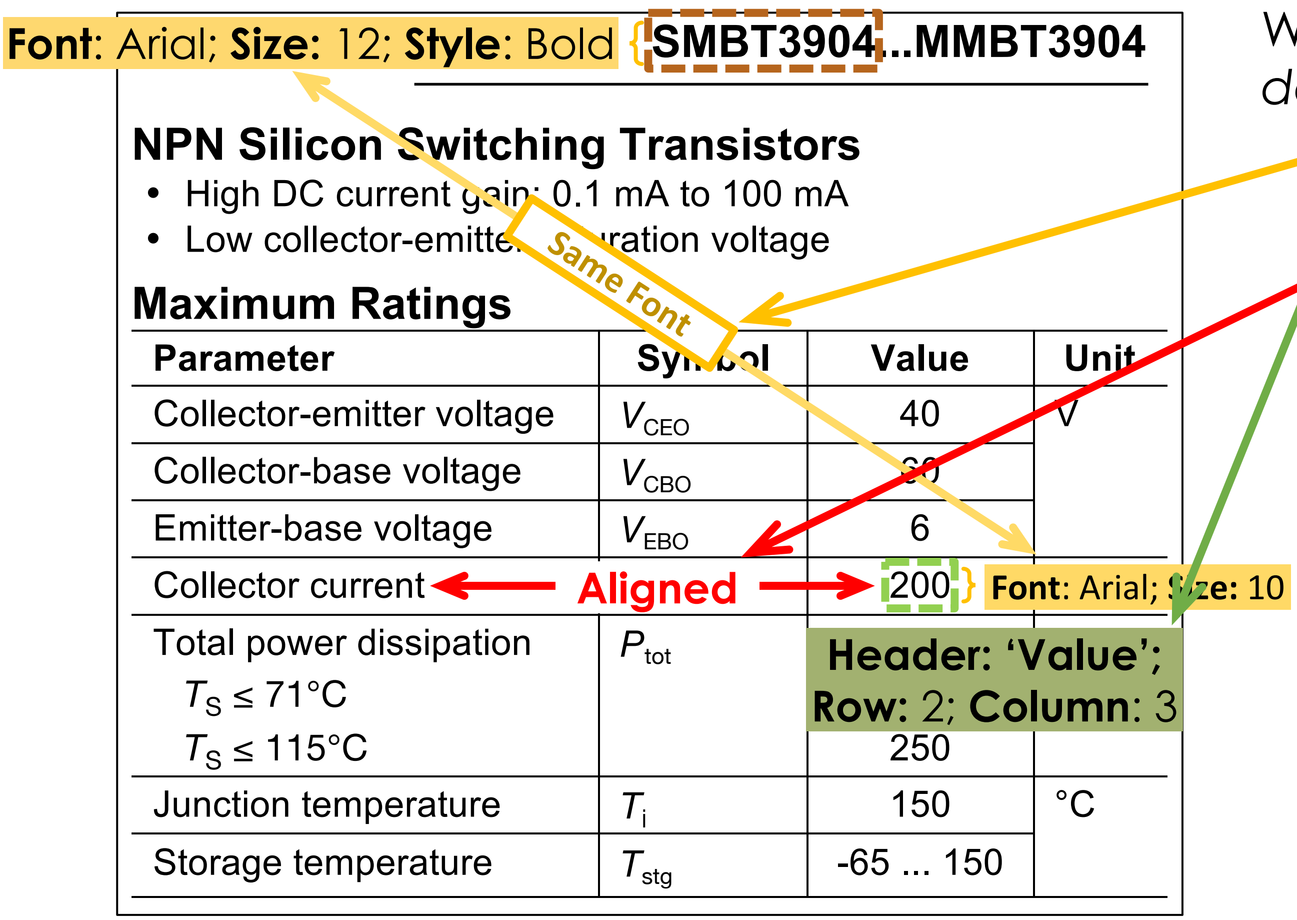
LSTM excels at relation extraction from text

Xu et al., 2015; Miwa et al., 2016; Zhang et al., 2016



Problem: LSTM networks struggle to capture the multimodal characteristics of richly formatted data.

Augmenting LSTM with Multimodal Features



Transistor Datasheet

SMBT3904...MMBT3904

NPN Silicon Switching Transistors

- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current		200	
Total power dissipation $T_S \leq 71°C$ $T_S \leq 115°C$	P_{tot}	 250	
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

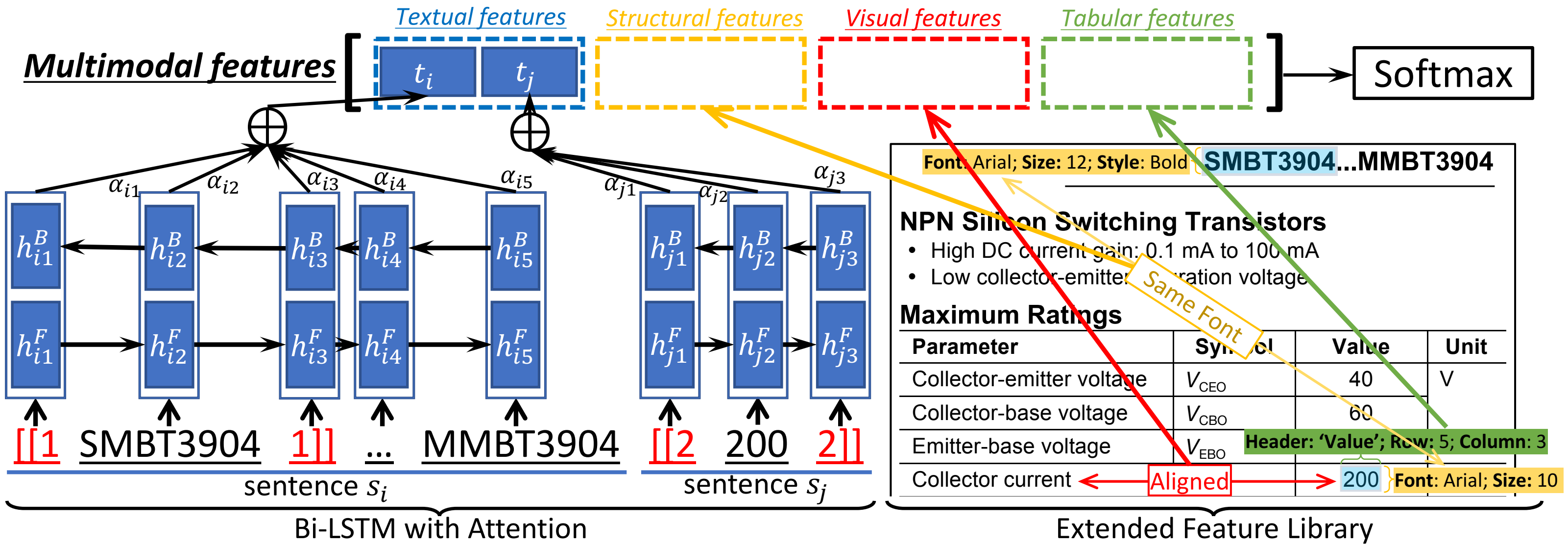
We use the multimodal information stored in the *document* to extract basic multimodal features:

- ❑ Structural features
- ❑ Tabular features
- ❑ Visual features

Augmentation with multimodal features captures signals a traditional LSTM would miss.

Fonduer's Multimodal LSTM

Signals from different modalities can be useful to find the information.

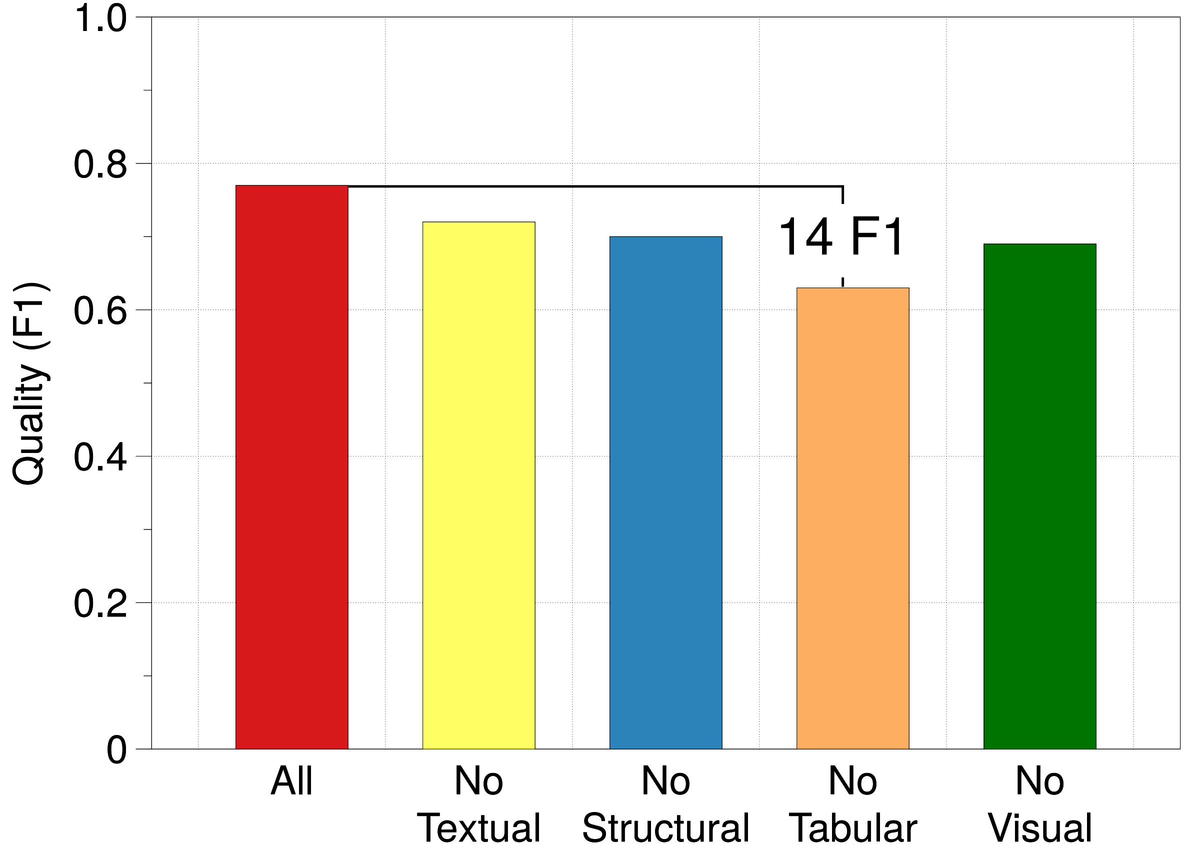


Fonduer: a KBC system that takes advantage of both techniques to reason about all available signals.

The impact of multimodal features

FONDUER

For transistor datasheets…



Multimodal features significantly impact the quality of extraction

Fonduer in the wild

Empirical results & real-world uses

Fonduer vs. Human-curated Knowledge Bases

GWAS Catalog	**Fonduer**
Same set of documents	
<u>Human</u>-created	<u>Machine</u>-created
10 years	**<6** months
1.0x extractions	**1.59x** extractions
	Precision **0.89**

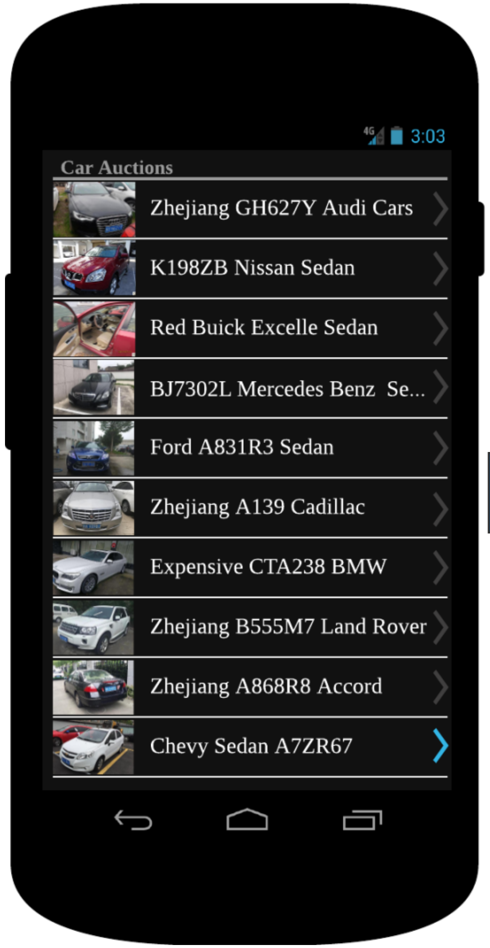
How people use Fonduer in industry

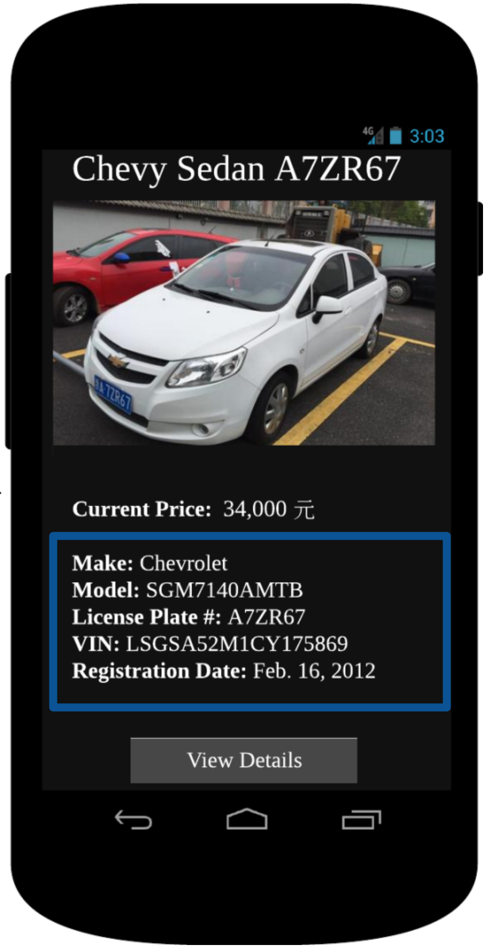
Input: User-customized HTML auction pages → **Output:** Structured knowledge base

Extract key facts (make, model, license, etc.)

Improve auction search quality and UX

Fonduer





Alibaba.com

Knowledge Base Construction from Richly Formatted Data

- Fonduer helps build high-quality KBC from richly formatted data
- Allows users to leverage multimodal signals
- Augments deep learning model with features from each data modality to achieve high quality
- Fonduer is supporting real world applications

Thank you!
Sen Wu
(senwu@cs.stanford.edu)

https://github.com/HazyResearch/fonduer

The Fonduer data model

Richly formatted data

SMBT3904...MMBT3904

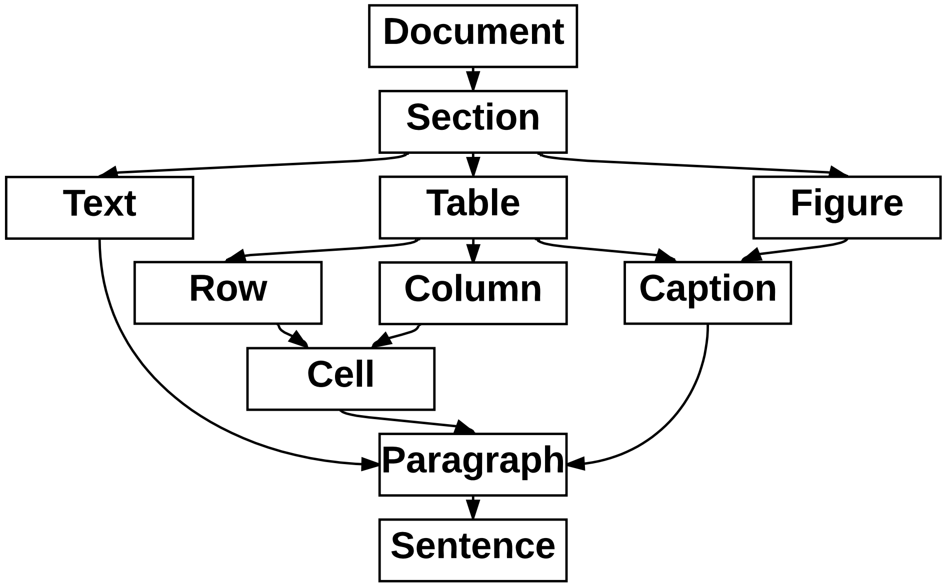
NPN Silicon Switching Transistors

- High DC current gain: 0.1 mA to 100 mA
- Low collector-emitter saturation voltage

Maximum Ratings

Parameter	Symbol	Value	Unit
Collector-emitter voltage	V_{CEO}	40	V
Collector-base voltage	V_{CBO}	60	
Emitter-base voltage	V_{EBO}	6	
Collector current	I_C	200	mA
Total power dissipation $T_S \leq 71°C$ $T_S \leq 115°C$	P_{tot}	 330 250	mV
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-65 ... 150	

Data model



Fonduer automatically parses the richly formatted data into the data model that:

- ❑ Preserves structure/semantics across modalities
- ❑ Unifies a diverse variety of formats and styles
- ❑ Serves as the formal representation in KBC

Data cleaning

We want to detect and repair errors in a dataset

University of Chicago, *Cicago*, IL

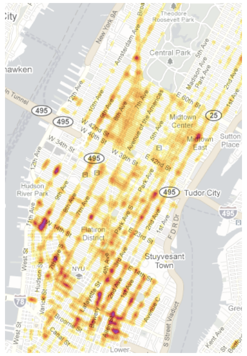
Where does data cleaning come up? All analytics!

THOMSON REUTERS

Data feeds

TWO SIGMA

Investment



Urban data

A simple example

Chicago's food inspection dataset

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Conflicts

Does not obey data distribution

Conflict

Detect and ***repair*** errors in a structured dataset

Constraints and minimality

Functional dependencies

c1: DBAName $\rightarrow$ Zip

c2: Zip $\rightarrow$ City, State

c3: City, State, Address $\rightarrow$ Zip

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Bohannon et al., 2005, 2007; Kolahi and Lakshmanan , 2005; Bertossi et al., 2011; Chu et al., 2013; 2015 Fagin et al., 2015

Constraints and minimality

Functional dependencies

c1: DBAName $\rightarrow$ Zip

c2: Zip $\rightarrow$ City, State

c3: City, State, Address $\rightarrow$ Zip

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Action: Fewer erroneous than correct cells; perform minimum number of changes to satisfy all constraints

Constraints and minimality

Functional dependencies

c1: DBAName $\rightarrow$ Zip

c2: Zip $\rightarrow$ City, State

c3: City, State, Address $\rightarrow$ Zip

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Error; correct zip code is 60608

Does not fix errors and introduces new ones.

External information

Matching dependencies

m1: $Zip = Ext\_Zip \rightarrow City = Ext\_City$

m2: $Zip = Ext\_Zip \rightarrow State = Ext\_State$

m3: $City = Ext\_City \wedge State = Ext\_State \wedge$

$\wedge\ Address = Ext\_Address \rightarrow Zip = Ext\_Zip$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Fan et al., 2009; Bertossi et al., 2010; Chu et al., 2015

External information

Matching dependencies

m1: $\text{Zip} = \text{Ext\_Zip} \rightarrow \text{City} = \text{Ext\_City}$

m2: $\text{Zip} = \text{Ext\_Zip} \rightarrow \text{State} = \text{Ext\_State}$

m3: $\text{City} = \text{Ext\_City} \wedge \text{State} = \text{Ext\_State} \wedge$
$\wedge \text{Address} = \text{Ext\_Address} \rightarrow \text{Zip} = \text{Ext\_Zip}$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t4	**Johnnyo's**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Action: Map external information to input dataset using matching dependencies and repair disagreements

External information

Matching dependencies

$$\text{m1: } \text{Zip} = \text{Ext\_Zip} \rightarrow \text{City} = \text{Ext\_City}$$
$$\text{m2: } \text{Zip} = \text{Ext\_Zip} \rightarrow \text{State} = \text{Ext\_State}$$
$$\text{m3: } \text{City} = \text{Ext\_City} \land \text{State} = \text{Ext\_State} \land$$
$$\land\ \text{Address} = \text{Ext\_Address} \rightarrow \text{Zip} = \text{Ext\_Zip}$$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t4	**Johnnyo's**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

External dictionaries may have limited coverage or not exist altogether

Quantitative statistics

Reason about co-occurrence of values across cells in a tuple

Estimate the distribution governing each attribute

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Example: Chicago co-occurs with IL

Hellerstein, 2008; Mayfield et al., 2010; Yakout et al., 2013

Quantitative statistics

Reason about co-occurrence of values across cells in a tuple

Estimate the distribution governing each attribute

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	**John Veliotis Sr.**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Again, fails to repair the wrong zip code

Let’s combine everything

Constraints and minimality

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo’s	Johnnyo’s	3465 S Morgan ST	Cicago	IL	60608

External data

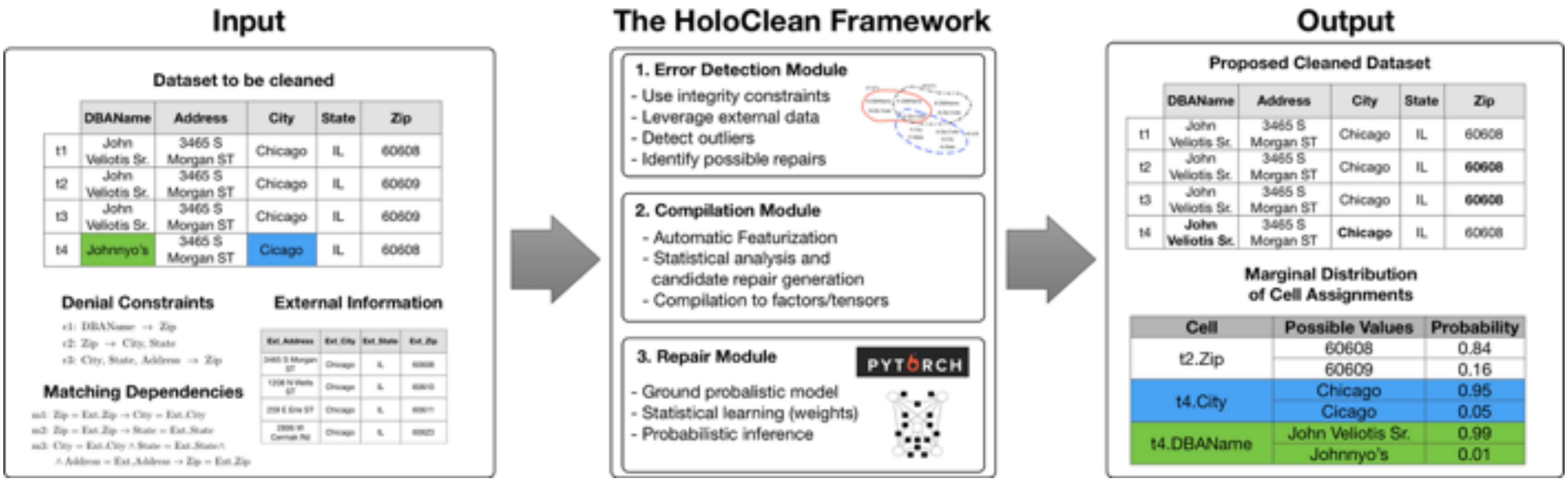
	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	**60608**
t4	Johnnyo’s	Johnnyo’s	3465 S Morgan ST	**Chicago**	IL	60608

Quantitative statistics

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo’s	3465 S Morgan ST	Chicago	IL	60609
t4	**John Veliotis Sr.**	Johnnyo’s	3465 S Morgan ST	**Chicago**	IL	60608

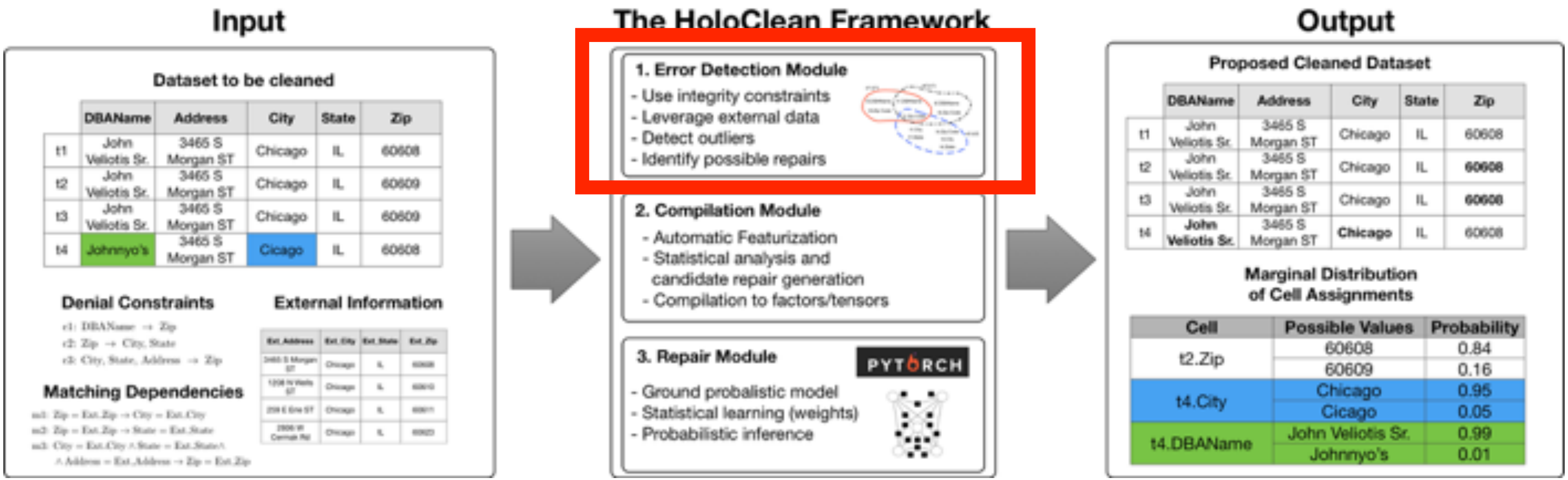
Different solutions suggest different repairs

Probabilistic data repairs



HoloClean
[VLDB'17]

Probabilistic data repairs



HoloClean
[VLDB'17]

Error detection in HoloClean

HoloClean focuses on repairing. Error detection is treated as black-b

Input

	Address	City	State	Zip
t1	3465 S Morgan ST	Chicago	IL	60608
t2	3465 S Morgan ST	Chicago	IL	60609
t3	3465 S Morgan ST	Chicago	IL	60609
t4	3465 S Morgan ST	Cicago	IL	60608

Error Detection

Example:

$$\mathrm{Zip} \rightarrow \mathrm{City}$$

External:

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

Output

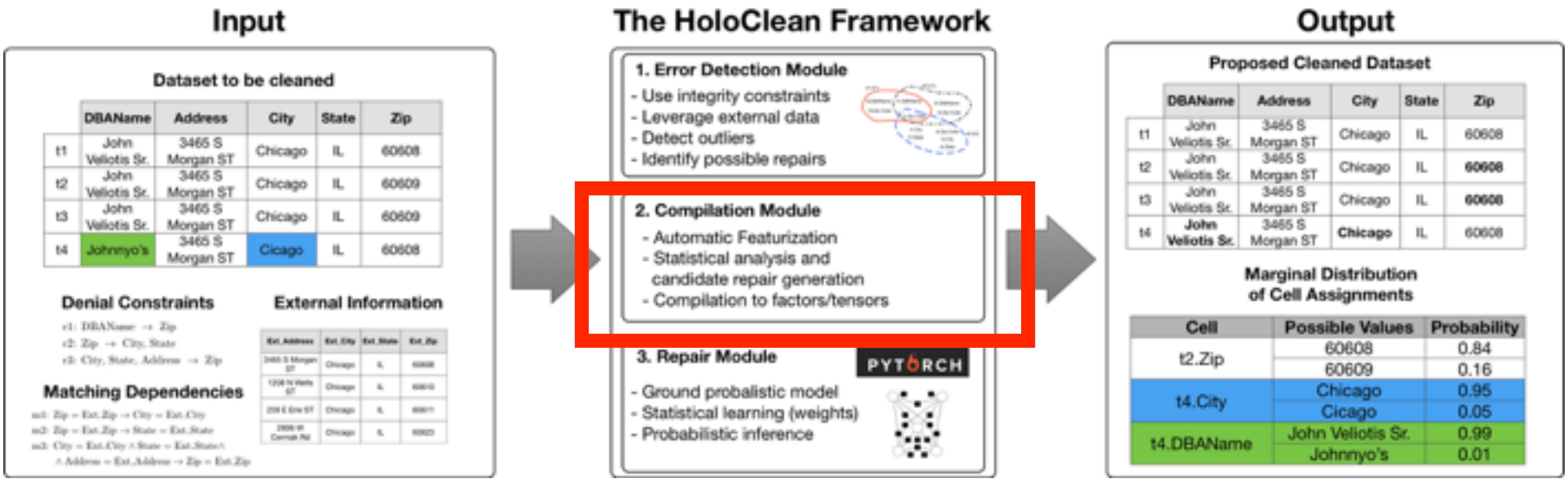
	Address	City	State	Zip
t1	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	3465 S Morgan ST	Chicago	IL	***60609***
t3	3465 S Morgan ST	Chicago	IL	***60609***
t4	3465 S Morgan ST	***Cicago***	IL	***60608***

Error detection splits input into ***correct*** *and* ***potentially erroneous*** *cells.*

: Correct cells

: Potentially erroneous cell

Probabilistic data repairs



HoloClean
[VLDB'17]

HoloClean's model for data repairs

	Address	City	State	Zip
t1	3465 S Morgan ST	*Chicago*	IL	*60608*
t2	3465 S Morgan ST	Chicago	IL	*60609*
t3	3465 S Morgan ST	Chicago	IL	*60609*
t4	3465 S Morgan ST	*Cicago*	IL	*60608*

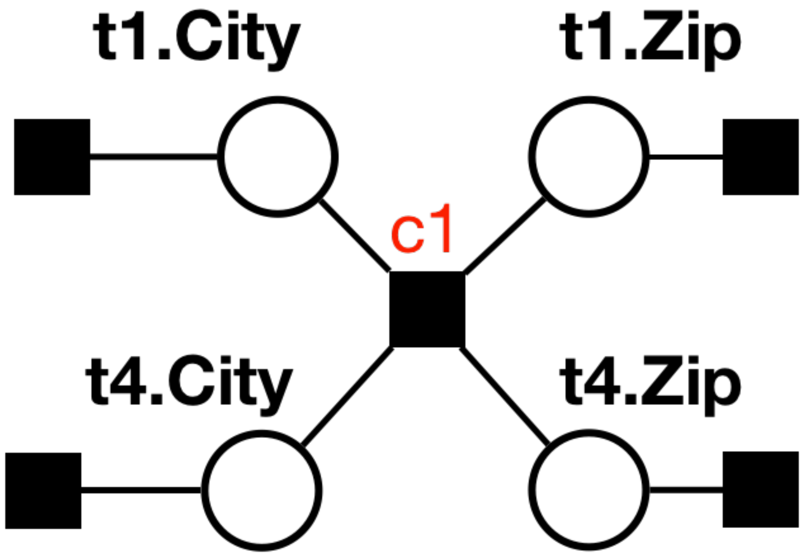
Each cell is a random variable

Value co-occurrences capture data statistics

Constraints introduce correlations

$$\mathrm{c1:\ Zip} \rightarrow \mathrm{City}$$

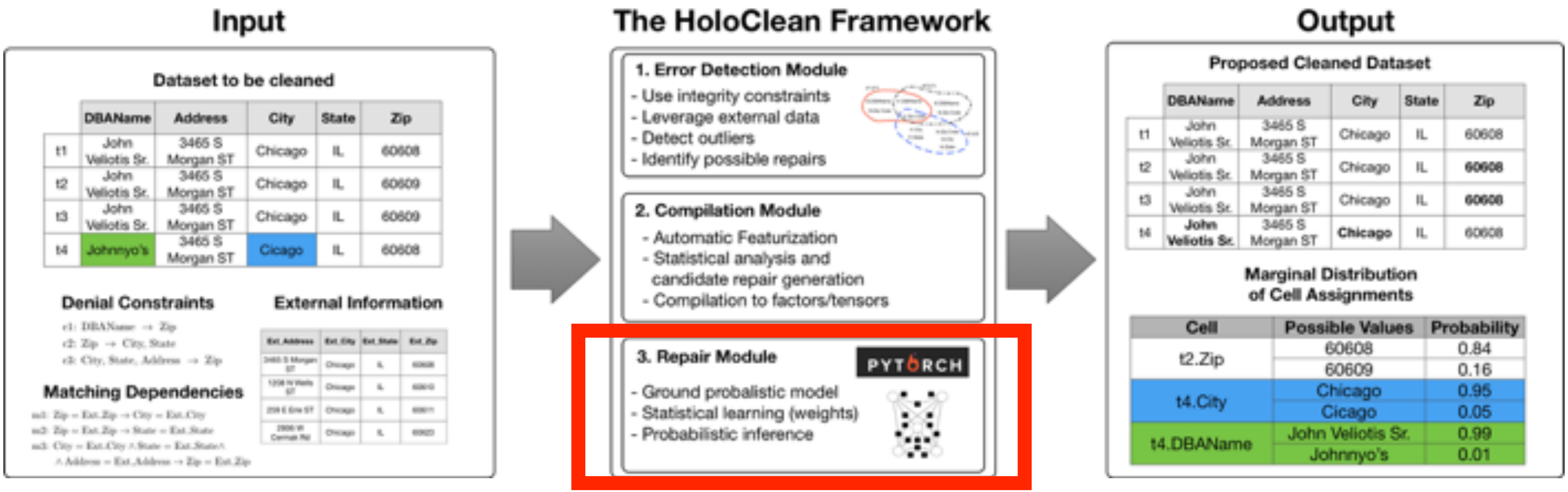
"Address= 3465 S Morgan St"



: Unknown (to be inferred) RV

: Factor (encodes correlations)

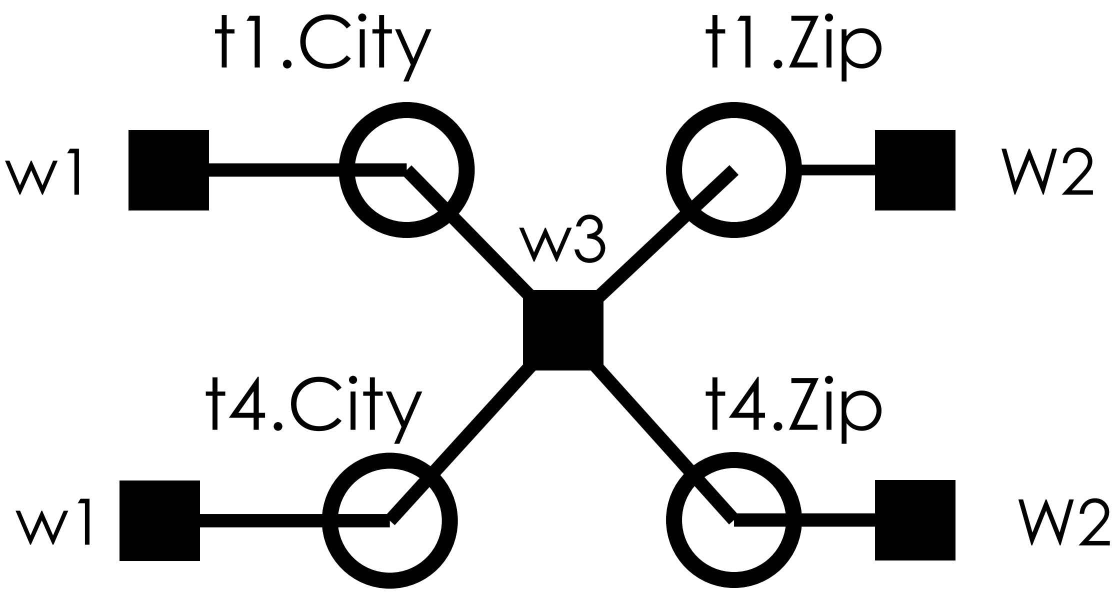
Probabilistic data repairs



HoloClean
[VLDB'17]

HoloClean's model

Factor Graph



Exponential family (canonical form)

$$\mathbf{w} = (w_1, w_2, \ldots, w_s)^T$$

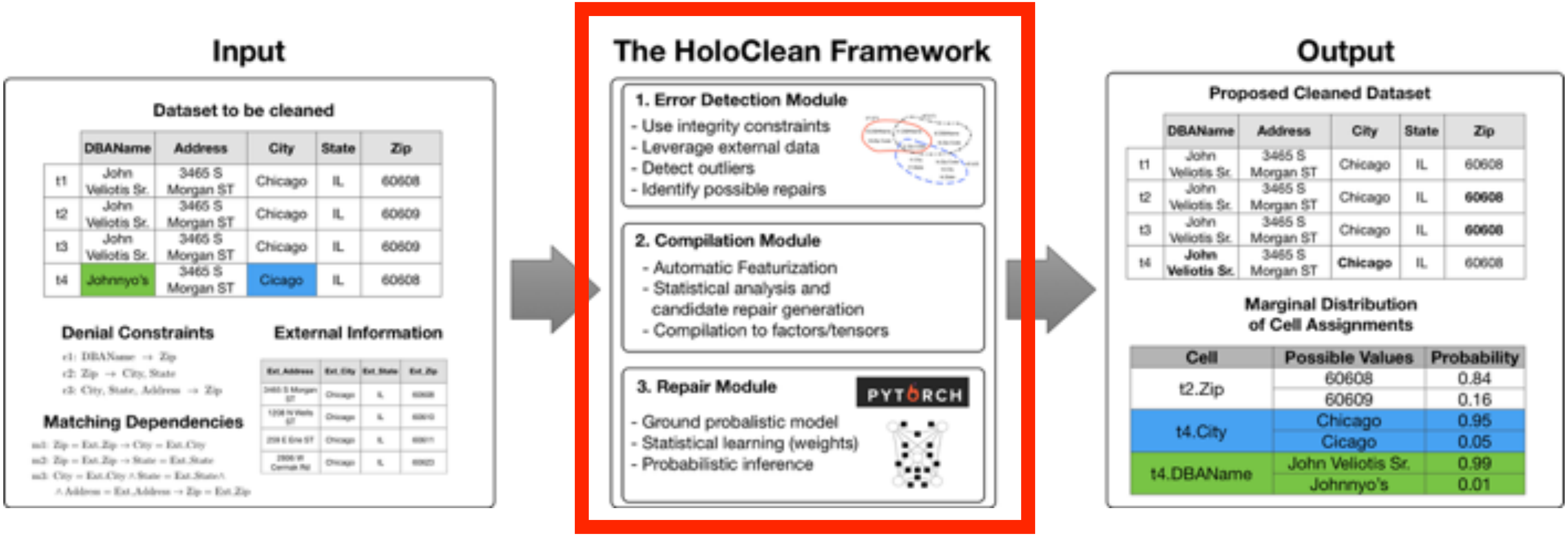
$$P(x|w) = \exp\left(\sum_{i=1}^{s} w_i T_i(x) - A(\mathbf{w})\right)$$

HoloClean automatically generates a factor graph that captures:

- Co-occurrences
- Correlations due to constraints
- Evidence due to external

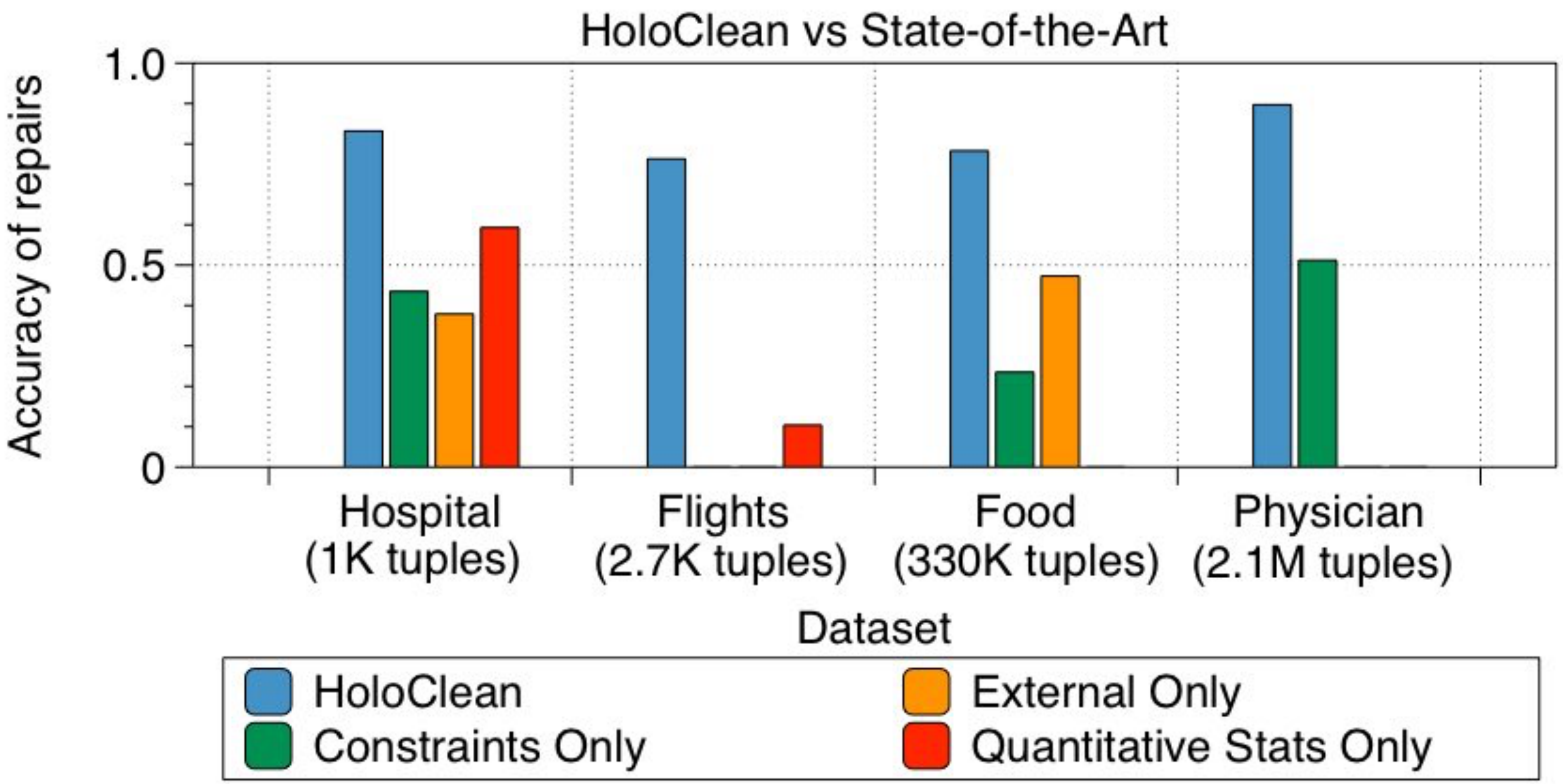
Repairing is a learning and inference problem:
Learn parameters w (use SGD) and infer the marginal distribution for unknown variables (use Gibbs sampling)

Probabilistic data repairs



HoloClean is a compiler for automatically generating probabilistic programs for data cleaning

HoloClean in practice



State-of-the-art does not scale or performs no correct repairs.

HoloClean: our approach combining all signals and using inference
Holistic[Chu,2013]: state-of-the-art for constraints & minimality
KATARA[Chu,2015]: state-of-the-art for external data
SCARE[Yakout,2013]: state-of-the-art ML & qualitative statistics

Scaling probabilistic inference

Challenge: Inference under constraints is #P-complete

Applying probabilistic inference naively does not scale to data cleaning instances with millions of tuples

Idea 1: Prune domain of random variables.

Idea 2: Relax constraints over sets of random variables to features over independent random variables.

Relaxing constraints

Tuple ID	University	State
t1	U of Chicago	IL
t2	U of Chicago	IL
t3	U of Chicago	CA

Functional dependency:

$$\text{University} \rightarrow \text{State}$$

"The same University must be in the same State"

FDs correspond to constraints over random variables (RVs)

Example:

$$\text{t1.University} = \text{t3.University} \implies \text{t1.State} = \text{t3.State}$$

Naive globally consistent model: It introduces correlations over four random variables.

We have D^4 possible worlds for such correlations.

D: domain of random variables

Relaxing constraints

Tuple ID	University	State
t1	U of Chicago	IL
t2	U of Chicago	IL
t3	U of Chicago	CA

Functional dependency: "The same University must be in the same State"

$$\text{University} \rightarrow \text{State}$$

Relax constraints to features over independent RVs (corresponds to a voting model)

Example:

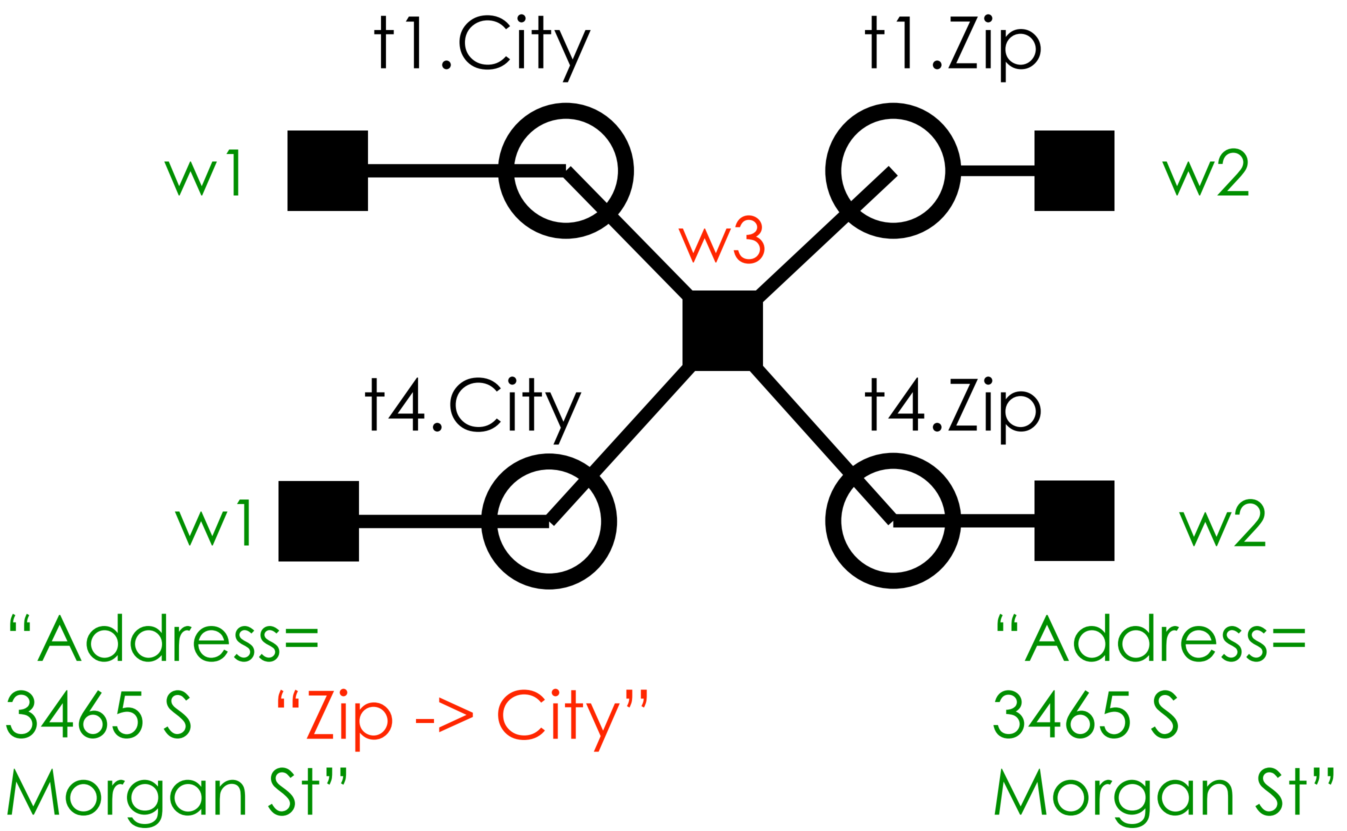
$$\text{t1.University} = \text{U of Chicago} \implies \text{IL} = \text{CA}$$
$$\text{U of Chicago} = \text{t3.University} \implies \text{IL} = \text{CA}$$
$$\text{U of Chicago} = \text{U of Chicago} \implies \text{t1.State} = \text{CA}$$
$$\text{U of Chicago} = \text{U of Chicago} \implies \text{IL} = \text{t3.State}$$

Only 4*D* possible worlds considered

HoloCleans' locally consistent model introduces features over independent random variables.

Relaxing constraints

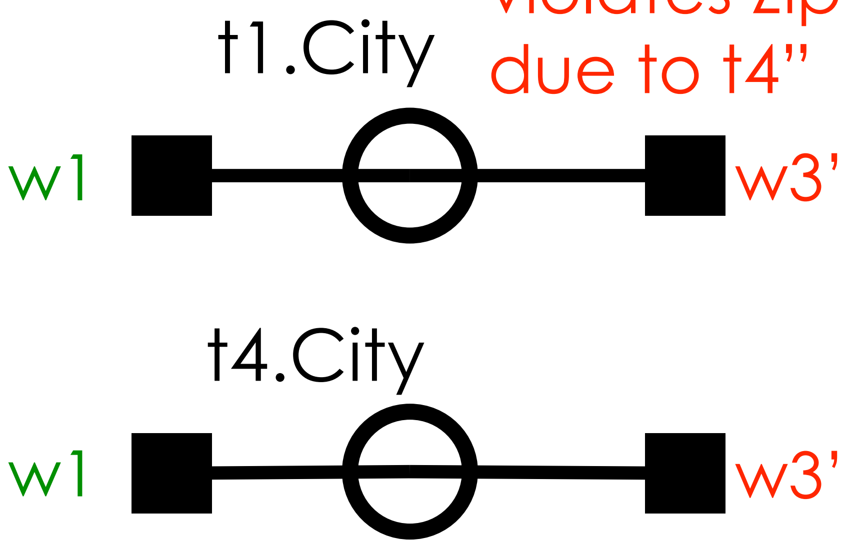
	Address	City	State	Zip
t1	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	3465 S Morgan ST	Chicago	IL	***60609***
t3	3465 S Morgan ST	Chicago	IL	***60609***
t4	3465 S Morgan ST	***Cicago***	IL	***60608***



Relaxing constraints

	Address	City	State	Zip
t1	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	3465 S Morgan ST	Chicago	IL	***60609***
t3	3465 S Morgan ST	Chicago	IL	***60609***
t4	3465 S Morgan ST	***Cicago***	IL	***60608***

"Assignment *Chicago* violates Zip -> City due to t4"



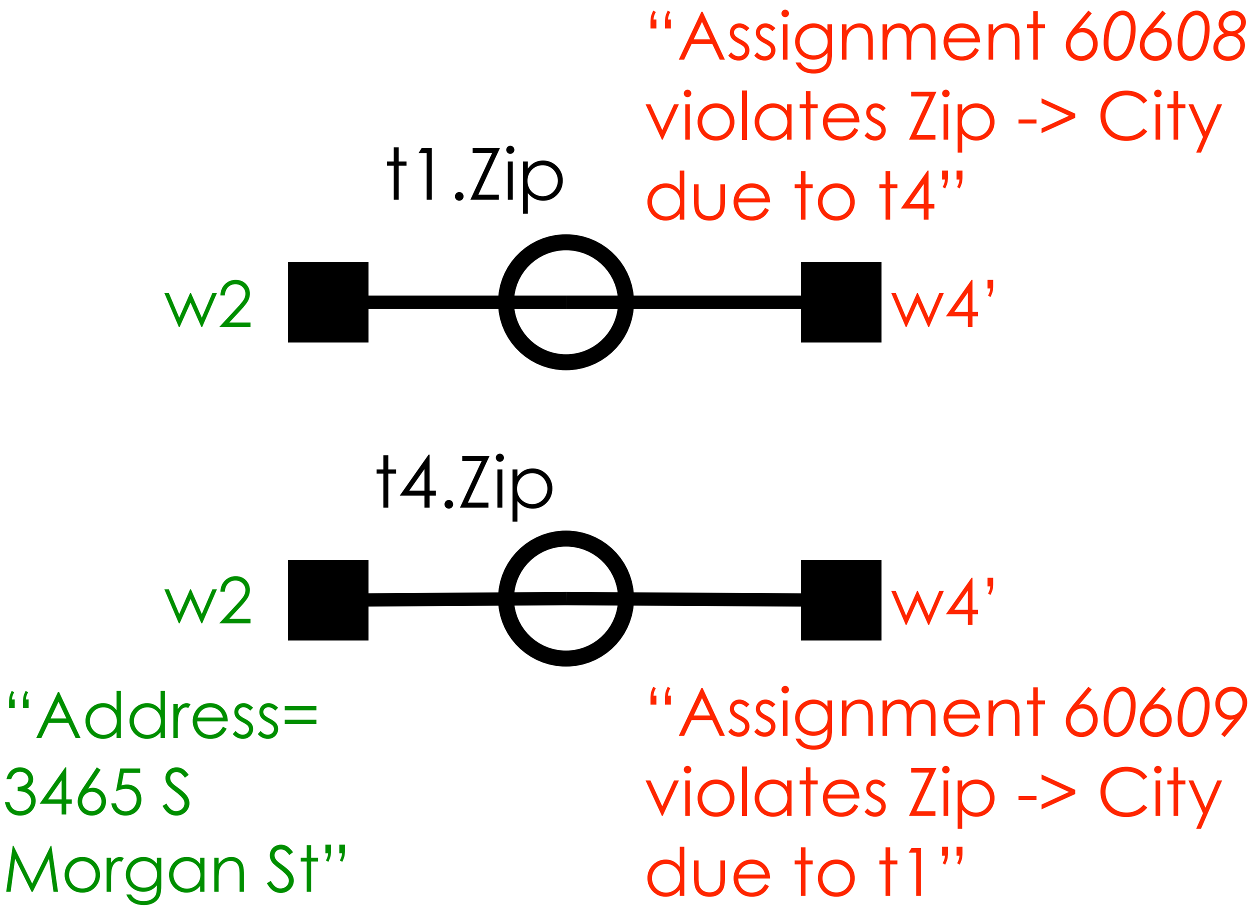
"Address= 3465 S Morgan St"

"Assignment *Cicago* violates Zip -> City due to t1"

We have one relaxed factor for each value in the domain of the RV

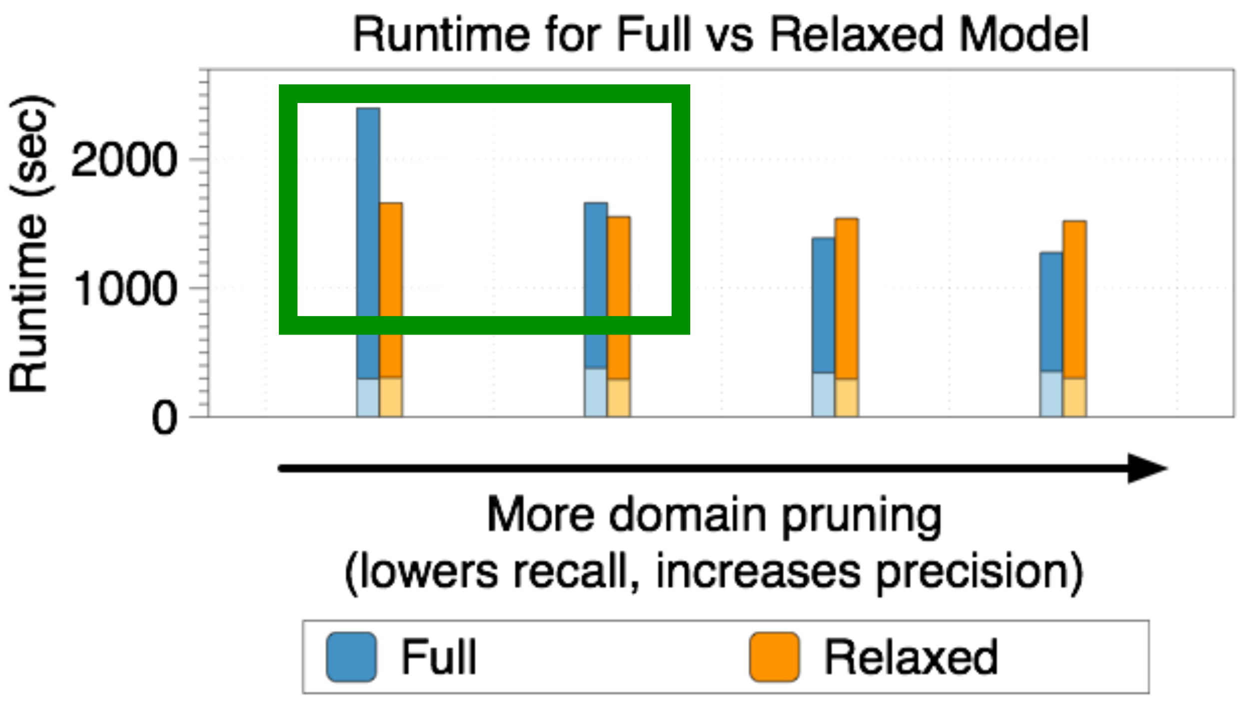
Relaxing constraints

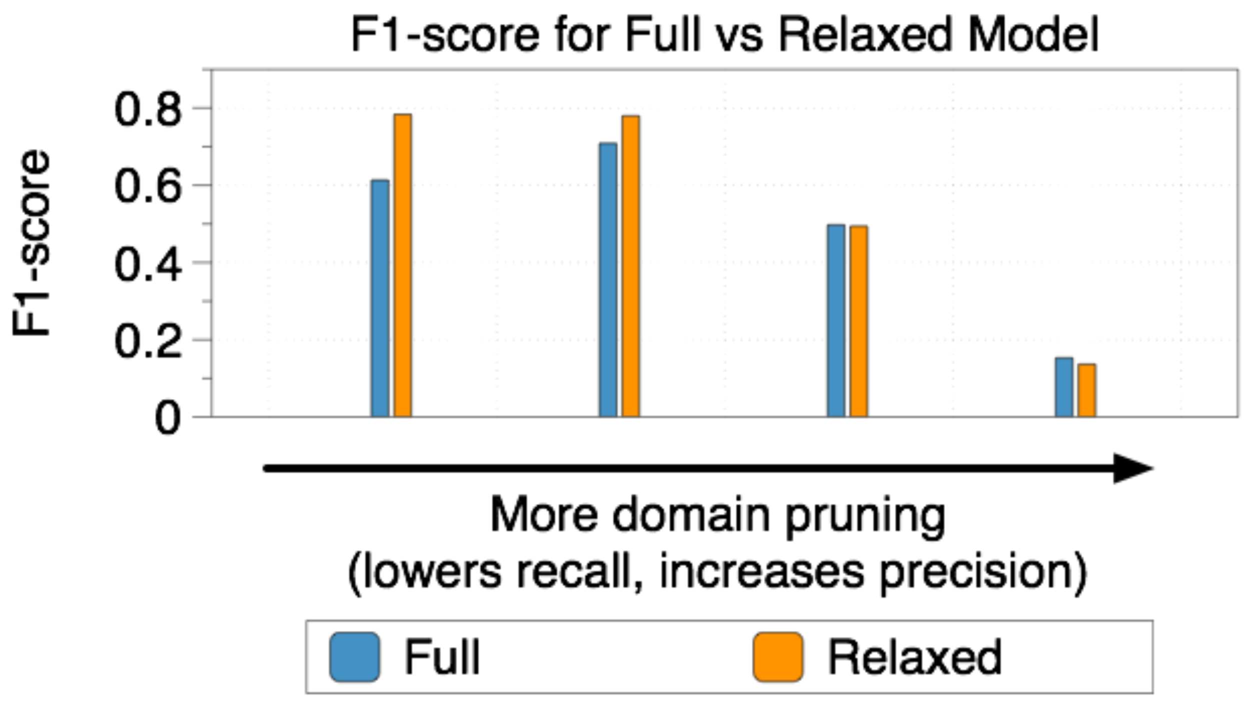
	Address	City	State	Zip
t1	3465 S Morgan ST	*Chicago*	IL	*60608*
t2	3465 S Morgan ST	Chicago	IL	*60609*
t3	3465 S Morgan ST	Chicago	IL	*60609*
t4	3465 S Morgan ST	*Cicago*	IL	*60608*



We have one relaxed factor for each value in the domain of the RV

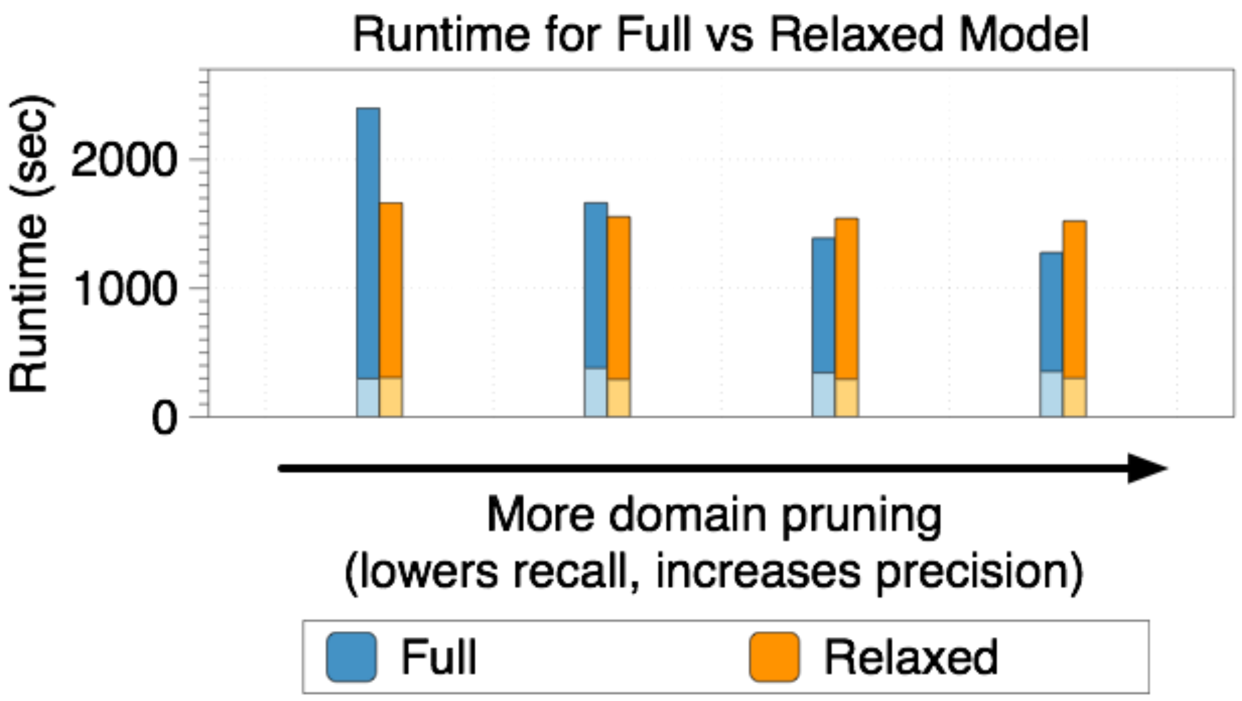
Relaxing constraints: In practice

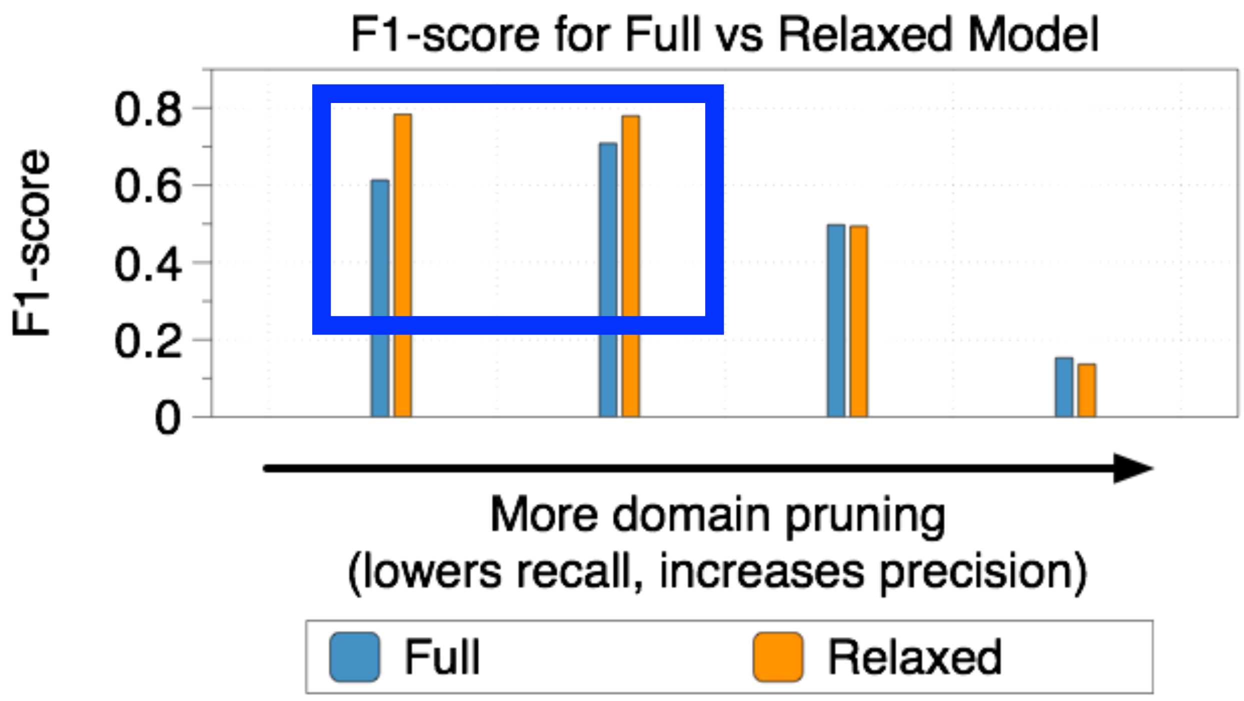




Faster compilation, learning, and inference when we do not prune the RV domain

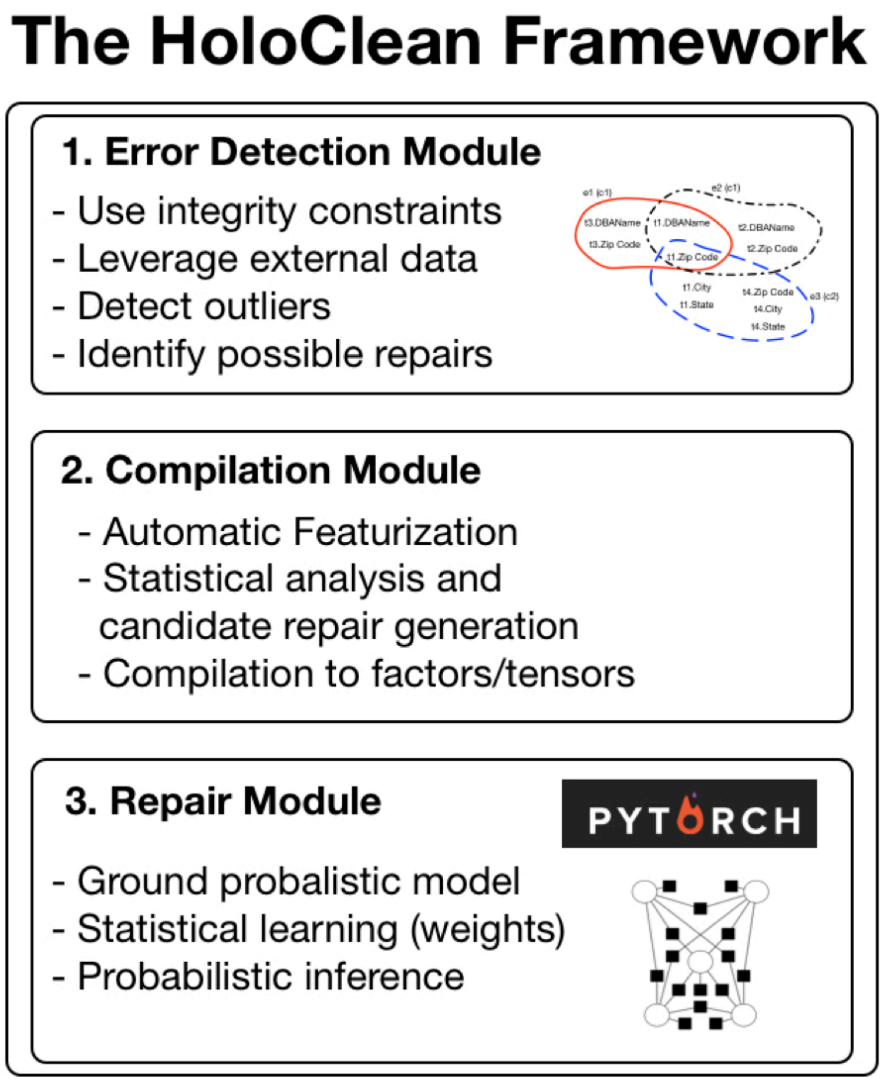
Relaxing constraints: In practice





Increased robustness (more accurate repairs) when RV domain is ill-specified (no heavy pruning used)

Data cleaning is a ML problem



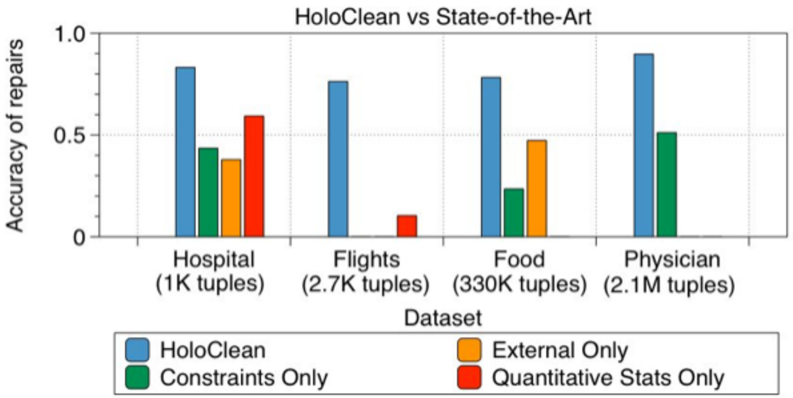
1. Combine disparate signals to perform accurate data repairs

2. Data cleaning is a statistical learning and inference problem
   - Transition from logic to probability

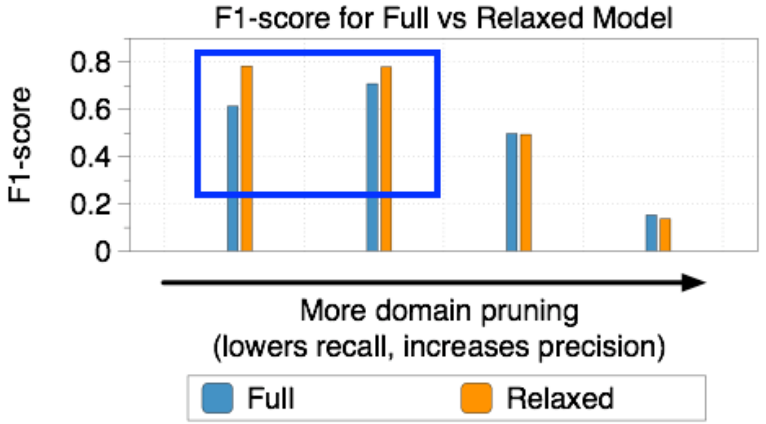
3. Connections to data vs knowledge tradeoffs in structured prediction

A quest for rigor

1. HoloClean provided empirical evidence the probabilistic methods work better



2. The ad-hoc relaxations for efficiency give more accurate data repairs



Why did logic fail us?
and *Why does relaxing constraints work?*

Back to the foundations: Logic and Databases

1. In 1969, Edgar F. Codd introduced the relational data model

2. In t007, C.J. Date wrote that logic and databases are "inextricably intertwined"

Two main uses of logic in databases

1. Logic is used as a database query language to express questions asked against databases.

2. Logic is used as a specification language to express integrity constraints in databases.

Noise models in DB theory

Coping with Inconsistent Databases

Two different approaches:

- Data Cleaning: Based on heuristics or specific domain knowledge, the inconsistent database is transformed to a consistent one by modifying (adding, deleting, updating) tuples in relations.
  - This is the main approach in industry (e.g., IBM InfoSphere Quality Stage, Microsoft DQS).
  - More engineering than science as quite often arbitrary choices have to be made.

- Database Repairs: A framework for coping with inconsistent databases in a principled way and without "cleaning" dirty data first.

Slide by Phokion Kolaitis
[SAT 2016]

Noise models in DB theory

Database Repairs

Definition (Arenas, Bertossi, Chomicki – 1999)

Σ a set of integrity constraints and I an inconsistent database.
A database J is a *repair* of I w.r.t. Σ if

- J is a consistent database (i.e., $J \models \Sigma$);
- J differs from I in a minimal way.

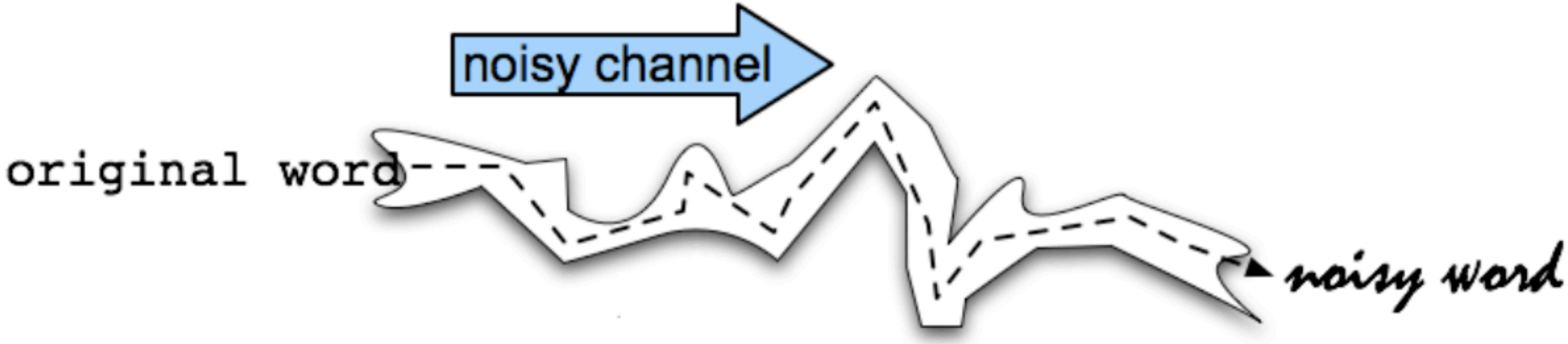
Fact

Several different types of repairs have been considered:

- Set-based repairs (subset, superset, $\oplus$-repairs).
- Cardinality-based repairs
- Attribute-based repairs
- Preferred repairs

Slide by Phokion Kolaitis
[SAT 2016]

Noise models outside DB



Noise models outside DB

Noisy Channel

1. We see an observation *x* in the noisy world

2. Find the correct world *w*

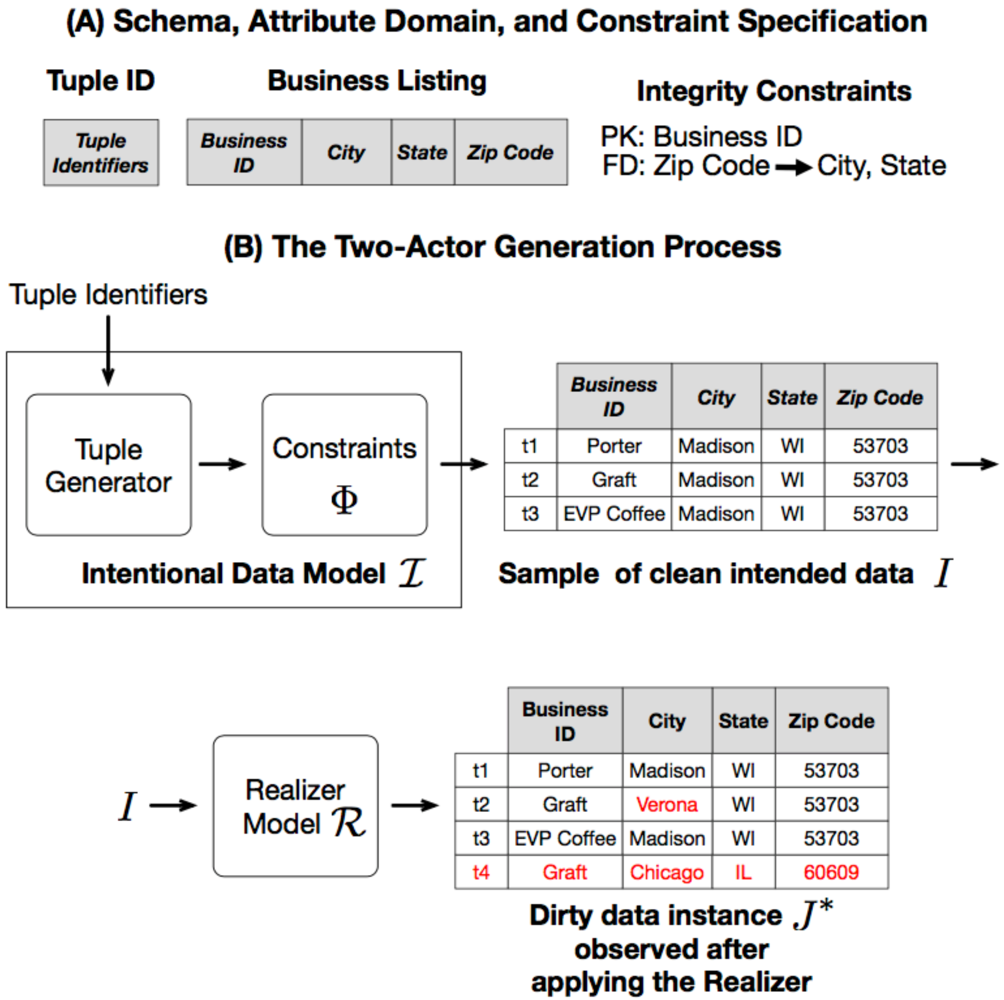
$$\hat{w} = \arg\max_{w \in W} P(w|x)$$

Applications

Speech, OCR, Spelling correction, Part of speech tagging, machine translations, etc…

Let's try new foundations for data cleaning!
…and see how they relate to logic.

Probabilistic Unclean Databases



[Work under submission, 2018]

Intentional Data Model

Step 1: Tuples are generated independently

$$\mathcal{P}(R^D) \stackrel{\text{def}}{=} \prod_{i \in ids_R(D)} (p_R \cdot \mathrm{TG}_R(D[i])) \times \prod_{\substack{i \in \varrho_R \setminus \\ ids_R(D)}} (1 - p_R).$$

Probability that tuple index was included in the world

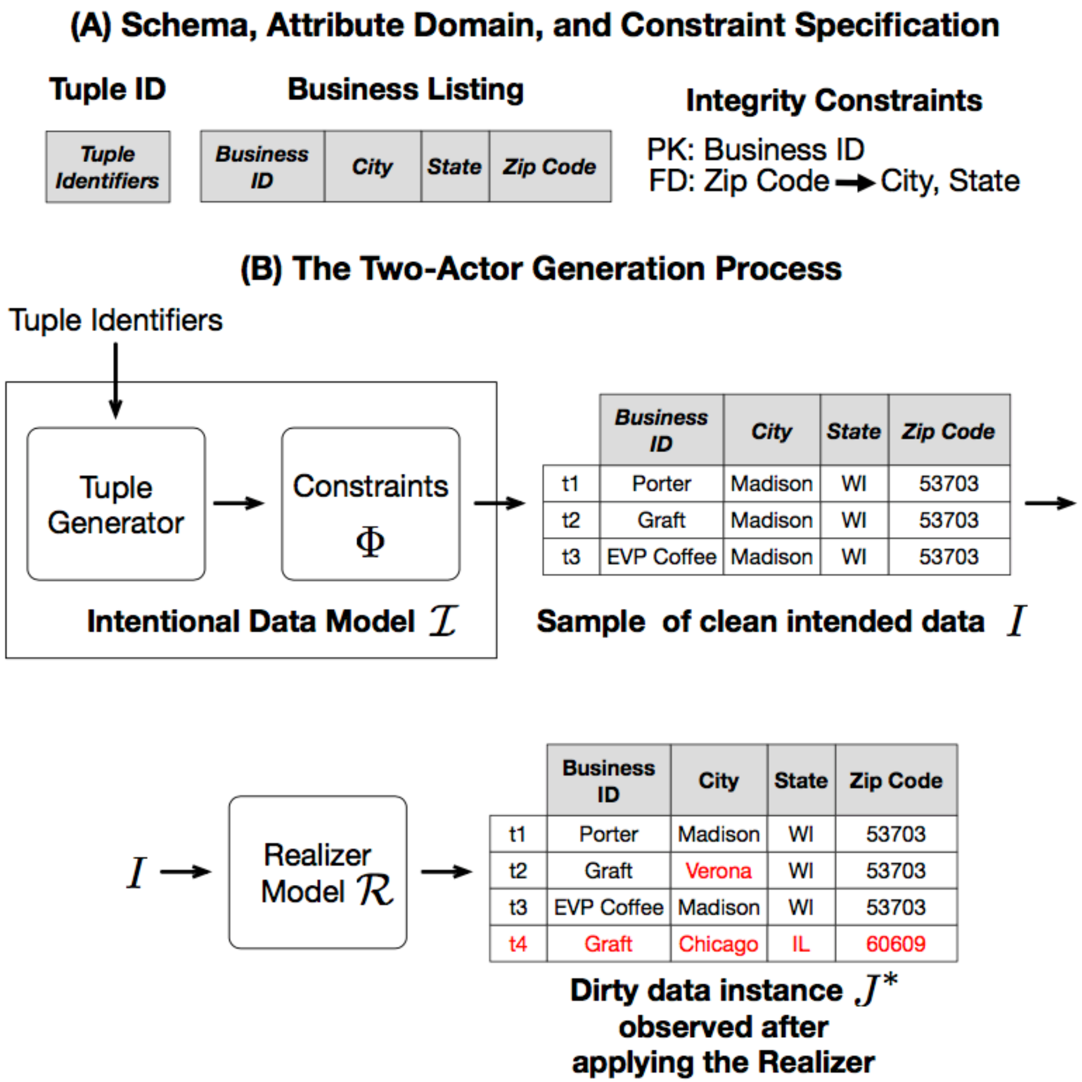
Probability obtaining a certain value

Step 2: Logical constraints ensure consistency

$$\mathcal{M}(D) \stackrel{\text{def}}{=} \frac{1}{Z} \times \mathcal{P}(D) \times \prod_{\varphi \in \Phi} e^{-w(\varphi)\cdot|V(D,\varphi)|}$$

Log-linear model penalizing invalid "possible worlds"

Probabilistic Unclean Databases



Realizer Model

$$\mathcal{R}_{\mathcal{I}}(I, J) \stackrel{\text{def}}{=} \mathcal{I}(I) \cdot \mathcal{R}_I(J)$$

Probability assigned to an intended instance *I*

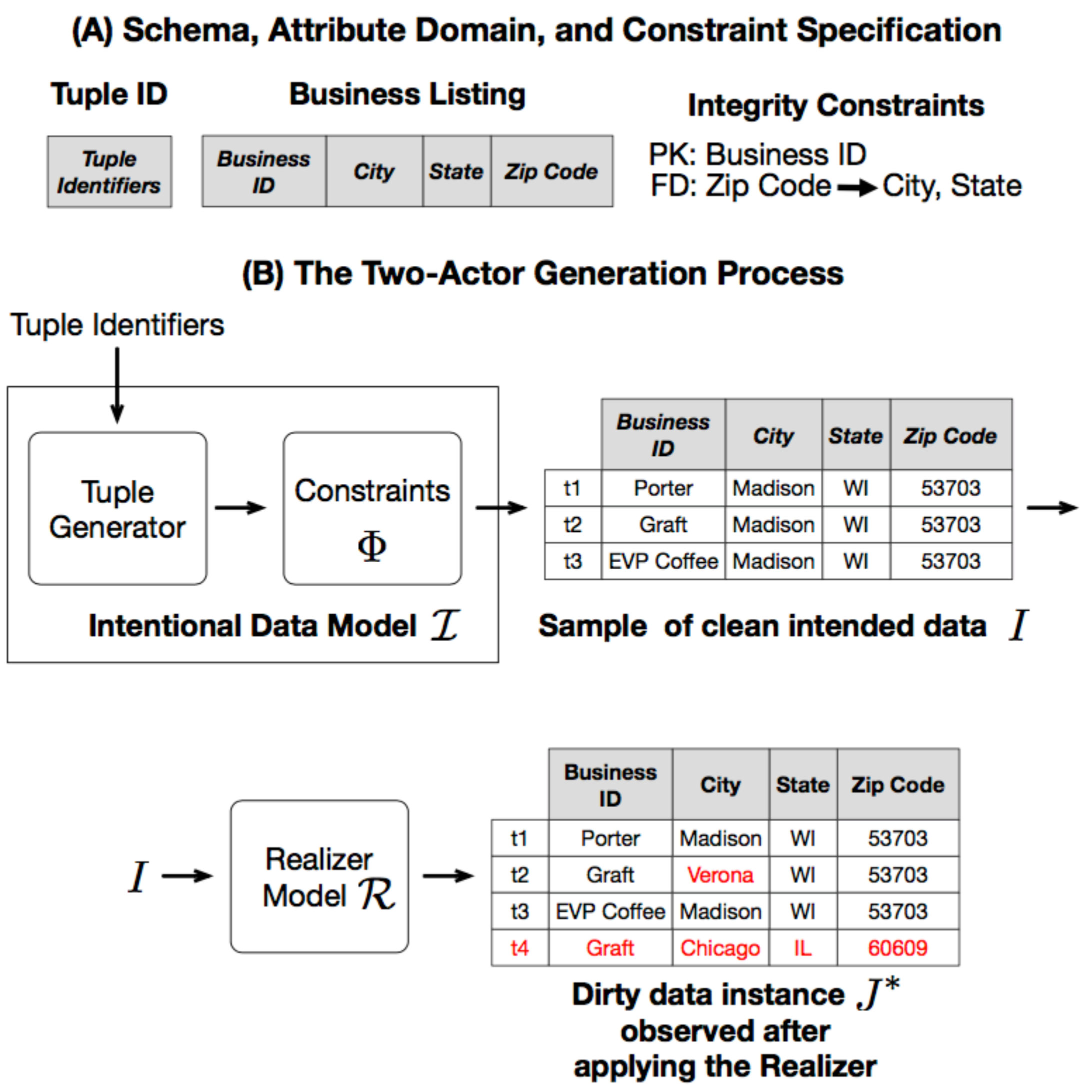
Conditional prob. of getting J given I

We consider two models

1. Insert unintended tuples (subset)
2. Update values of existing tuples

These models capture the data errors considered in prior works

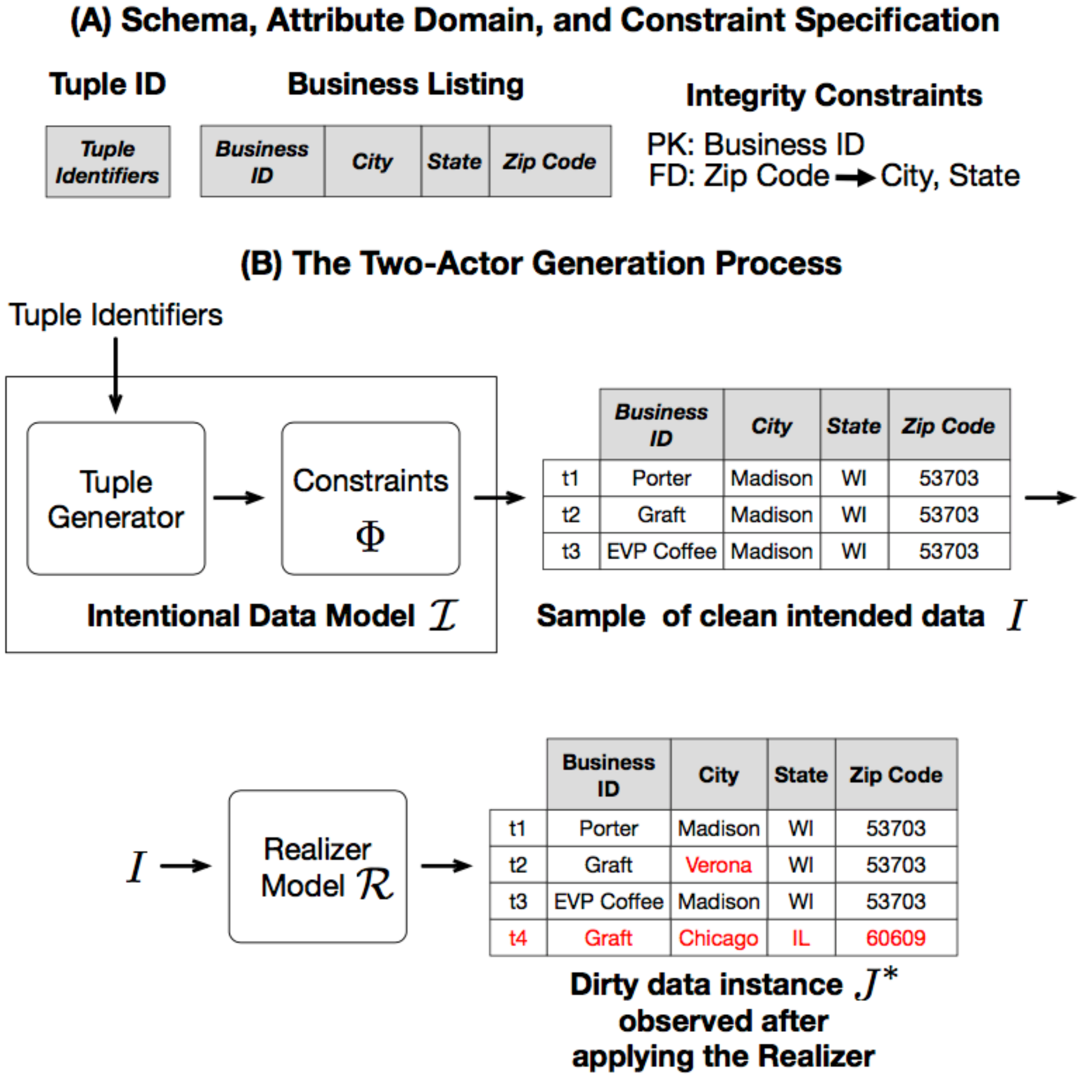
Probabilistic Unclean Databases



Computational problems

1. Cleaning: Find most probable ***I***
2. Probabilistic query answering (PQA): evaluate a query directly on ***J***
3. Learning Intentional and Realizer models

Probabilistic Unclean Databases



	Business ID	City	State	Zip Code
t1	Porter	Madison	WI	53703
t2	Graft	Madison	WI	53703
t3	EVP Coffee	Madison	WI	53703

	Business ID	City	State	Zip Code
t1	Porter	Madison	WI	53703
t2	Graft	Verona	WI	53703
t3	EVP Coffee	Madison	WI	53703
t4	Graft	Chicago	IL	60609

Preliminary Results

1. Cleaning: Connections to minimum repairs
2. Cleaning is in P-time for key constraints
3. Connections to consistent query answering
4. Learning from **one** noisy database without training data

Probabilistic Cleaning vs. Minimal Repairs

Theorem

For a ***subset realizer*** with ***low noise*** probabilistic repairs and minimal subset repairs are equivalent.

Subset realizer: Noisy channel that introduces new tuples
Low noise: probability of insertion from realizer lower than probability of insertion from intentional model

No assumptions on *tuple independence or attribute value independence.*

Probabilistic Cleaning vs. Minimal Repairs

Theorem

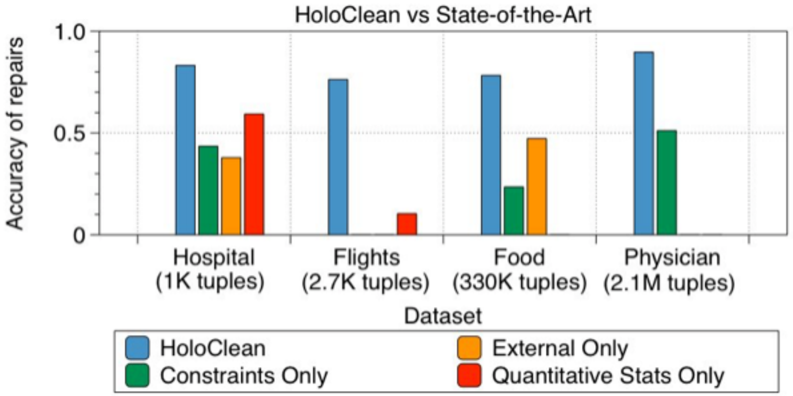
For an ***update realizer*** with ***low noise*** probabilistic repairs and cardinality minimal subset repairs are equivalent when *(1) tuples are independent* and *(2) tuple attribute assignments are independent!*

Update realizer: Noisy channel that permutes the values of cells (tuple attributes)
Low noise: probability of update less than 0.5

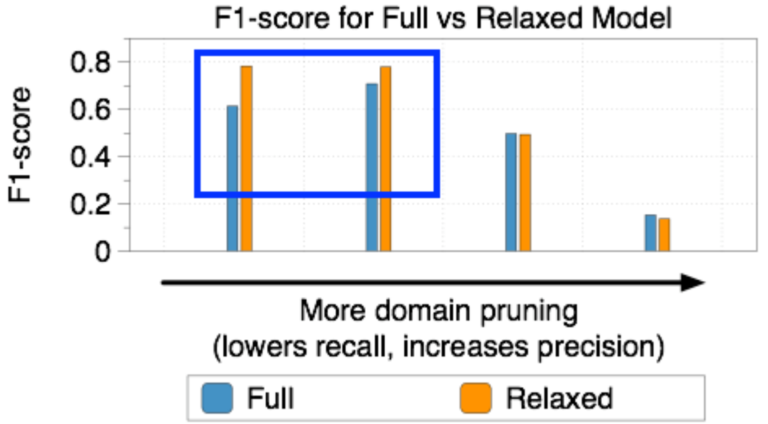
Strong assumptions that violate the relational model!

A quest for rigor

1. HoloClean provided empirical evidence the probabilistic methods work better



2. The ad-hoc relaxations for efficiency give more accurate data repairs



Why did logic fail us?
and *Why does relaxing constraints work?*

How hard is structured prediction?

Cleaning is a structured prediction problem

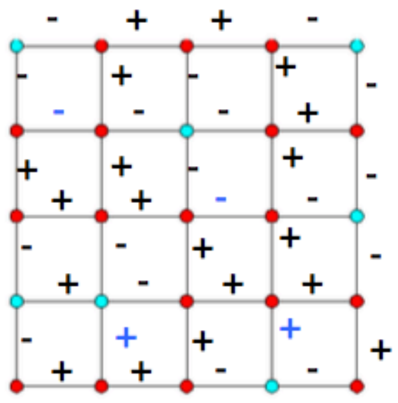
Our relaxation corresponds to an approximation for structured prediction

Recent work is targeting hardness of structured prediction

Globerson et al., ICML 2015
Foster et al., AISTATS 2018

Setup: (with noise)

- known graph G=(V,E)
- unknown labeling X:V -> {0,1}
- given noisy parity of each edge
  - flipped with probability p



Goal: (approximately) recover X.

Formally: want algorithm A: $\{+,-\}^E \to \{0,1\}^V$ that minimizes worst-case expected Hamming error:

$$\max_X \{E_{L\sim D(X)}[error(A(L),X)]\}$$

We are working on extensions to ***categorical variables*** and ***hypergraphs***.