Data Programming: Creating Large Training Sets, Quickly

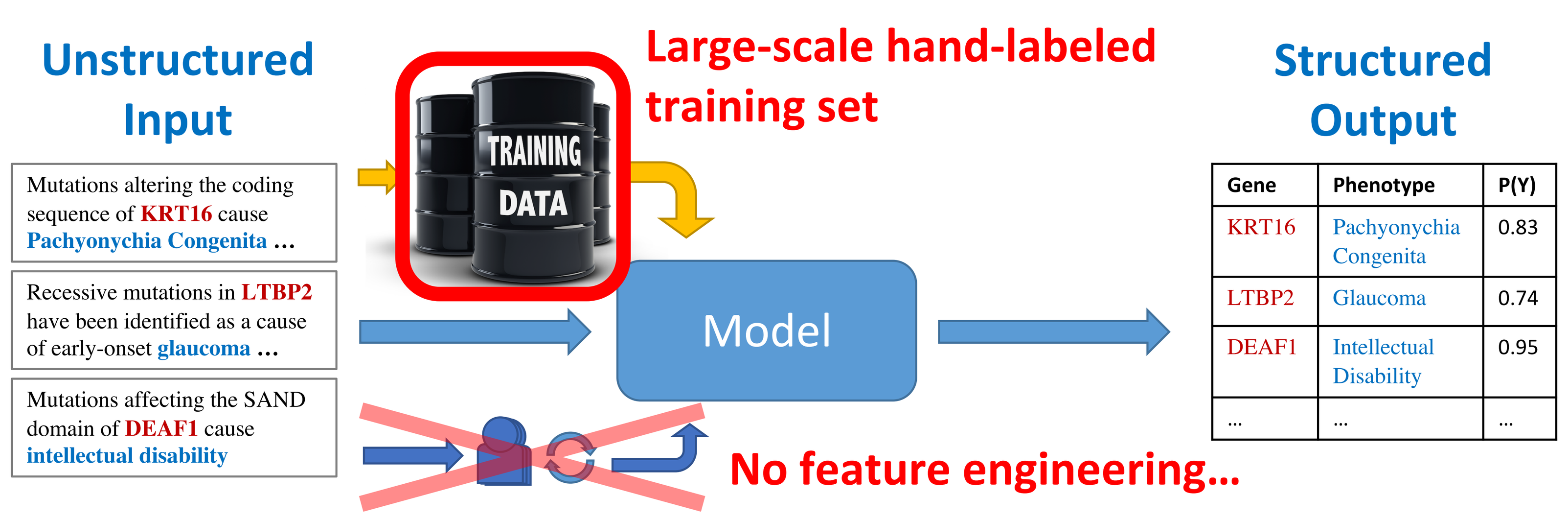
Alex Ratner (ajratner@stanford.edu), Chris De Sa, Sen Wu, Daniel Selsam, Chris Ré
Stanford University

Training Data: The *New* New Oil in Today's ML

Motivation:

- For many of today's models (e.g. deep learning), we **no longer need to do manual feature engineering!**
- ***However, these models require massive training sets…***
- ***Training set creation & management* is the key bottleneck in real-world applications!** The current way of hand-labeling data is prohibitively **expensive**, **slow**, and **brittle**

Example ML pipeline today (information extraction problem):

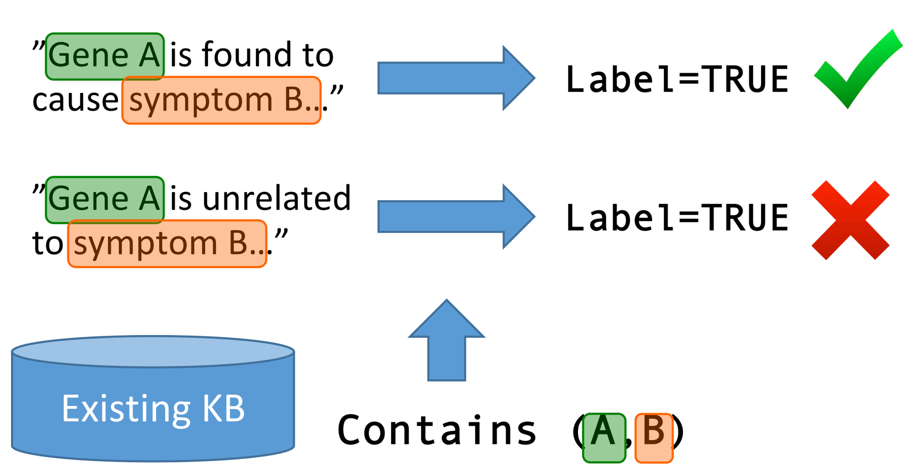


Generating Training Data Programmatically

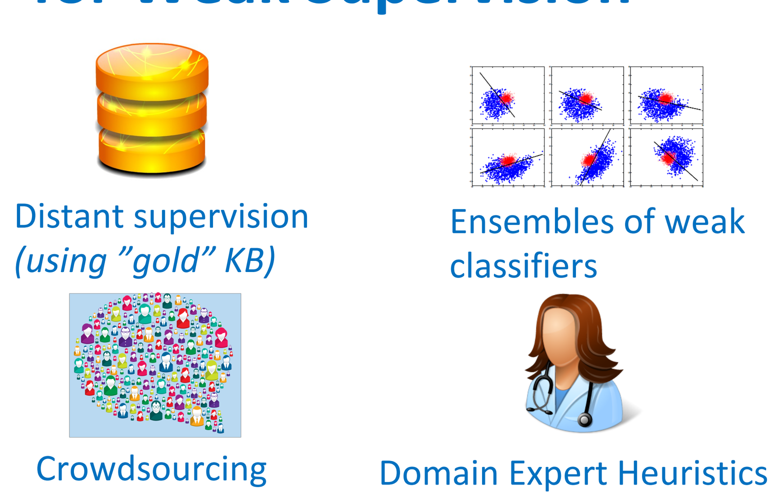
Creating Noisy Training Sets with Labeling Functions

In data programming, users write *labeling functions (LFs),* which are just scripts that noisily label subsets of the data. An example where we label relations in text based on an existing knowledgebase*:*

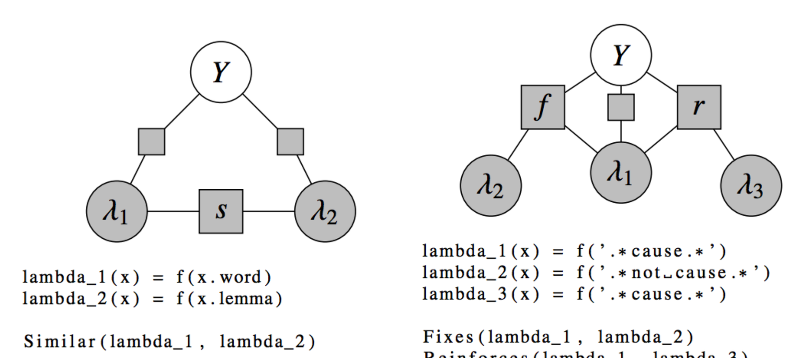
```
def lf1(x):
  cid = (x.gene, x.pheno)
  if cid in KB:
    return 1
  else:
    return 0
```



A Unifying Framework for Weak Supervision

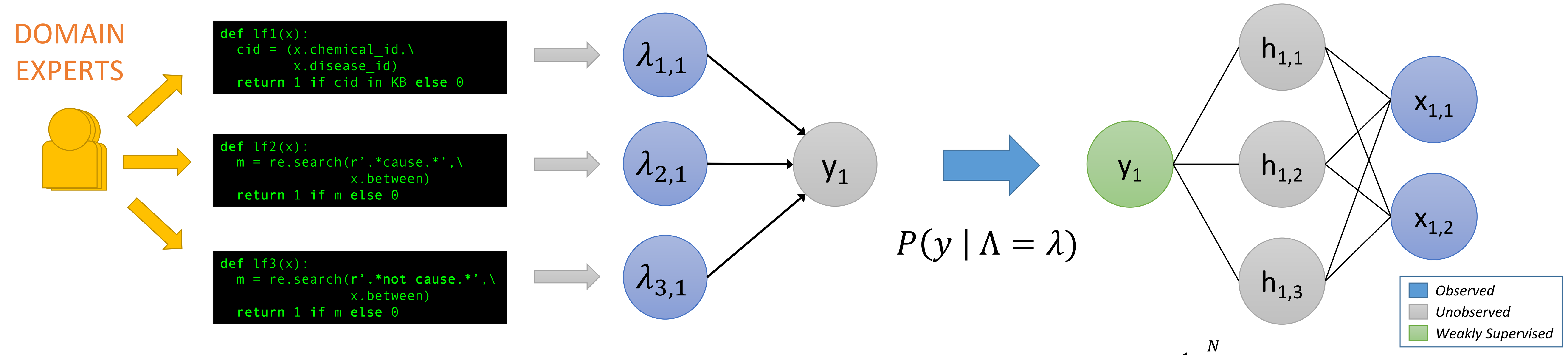


Dependencies Between LFs



We can also include dependencies between the LFs!

The Data Programming Pipeline: Modeling the Training Set Creation Process



$$\lambda_i : X \mapsto Y \cup \{0\}$$

$$\pi = P(y, \Lambda)$$

$$\hat{w} = \mathrm{argmin}_w \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{(y,\Lambda=\lambda)\sim\pi}\left[l\left(w, x^{(i)}, y^{(i)} = y\right)\right]$$

Labeling Functions (LFs)

In data programming, users focus on writing *labeling functions (LFs).* Labeling functions encode various heuristics or weak supervision signals to programmatically label training data.

Generative Model

The LFs define a generative model of the labeling process. By learning this model, based on the overlaps between LFs, we learn the accuracies of the LFs, and are able to denoise the training set they generate.

Noise-Aware Discriminative Model

We then train any discriminative model using a modified *noise-aware* loss function, which simply minimizes the expected loss with respect to the predictions of the generative model.

Theorem: Scaling with Unlabeled Data

Given a constant-order number of LFs, we get the same asymptotic scaling as in supervised methods—but ***with respect to unlabeled data!***

Theorem: Independent Case*

If:

1. π can be represented by our model family
2. Our noise-aware risk minimizer has bounded gen. risk
3. $y \perp f(x) \mid \lambda(x)$
4. We have a sufficient number of LFs with enough coverage & accuracy

Then: $\tilde{O}(\epsilon^{-2})$ ***unlabeled*** training points allow the algorithm to achieve $O(\epsilon)$ generalization risk (*using SGD + Gibbs sampling)*

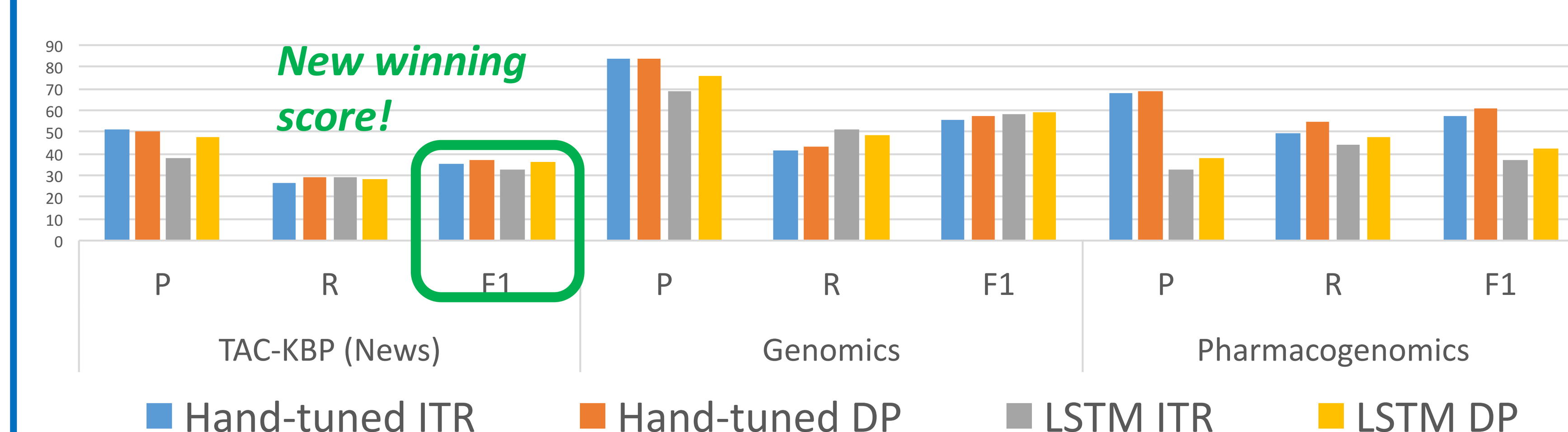
*For full theorem statement, more general form, and corresponding theorem for the case when LF dependencies are included, see paper

snorkel

We are implementing an easy-to-use information extraction framework, ***Snorkel***, using data programming *(snorkel.stanford.edu)*

Experimental Results: Information Extraction from Text

We test data programming (DP) on text information extraction problems, comparing to a distant supervision approach where rules to create training data were encoded as a simple *if-then-return (ITR)* script



We get significant improvements across a range of applications and LFs—notably, even more so with deep learning approaches!

Application	# of LFs	Coverage (%)	Training Set Size	Overlap (%)	Conflict (%)	F1 Improvement (Human Features)	F1 Improvement (LSTM)
TAC-KBP	40	29	2M	1.38	0.15	1.92	3.12
Genomics	146	54	256K	26.71	2.05	1.59	0.47
Pharma	7	8	129K	0.35	0.32	3.60	4.94