Rapid Biomedical Knowledge Base Construction from Unstructured Data

Snorkel Workshop

Jason Alan Fries, Postdoctoral Fellow

jfries@stanford.edu

Day 1: Agenda Outline

Designing Labeling Functions (LFs)

- Pattern Matching & Distant Supervision
- Evaluating LF Performance
- Snorkel API / Writing Labeling Functions

Generative Model: Unifying Supervision

- Simple Baseline: Majority Vote
- Automatically Learning LF Accuracies
- LF Dependency Learning

Discriminative Model: “Compiling” Rules into Features

- Training with Probabilistic Labels
- The Death of Manual Feature Engineering
- Why Do We Need the Discriminative Model?

Day 1: Agenda Outline

Application Development:
Introducing Schemas and Evaluation Plans

- Day 2 Preview: Designing a Good Evaluation Plan
- Schema Design Template

Welcome Reception

Terminology

Entity

Concepts that can be separated into meaningful categories

Person Barack Obama
Place Washington D.C.
Organization Apple, Inc.

Relation

Semantic associations between 2 or more entities

Spouse

Barack Obama
Person

Michelle Obama
Person

Knowledge Base

A repository for structured information

A network of all **chemical-induced disease relations** found in **PubMed**

Imagine for a moment …

Entertainment News Website

The entertainment news website TMZ wants **YOU** to build a state-of-the art text-mining system for tracking celebrity marriage gossip…

Being a top-notch (somewhat mercenary) data scientist…
You quickly recognize this as a **relation extraction task**

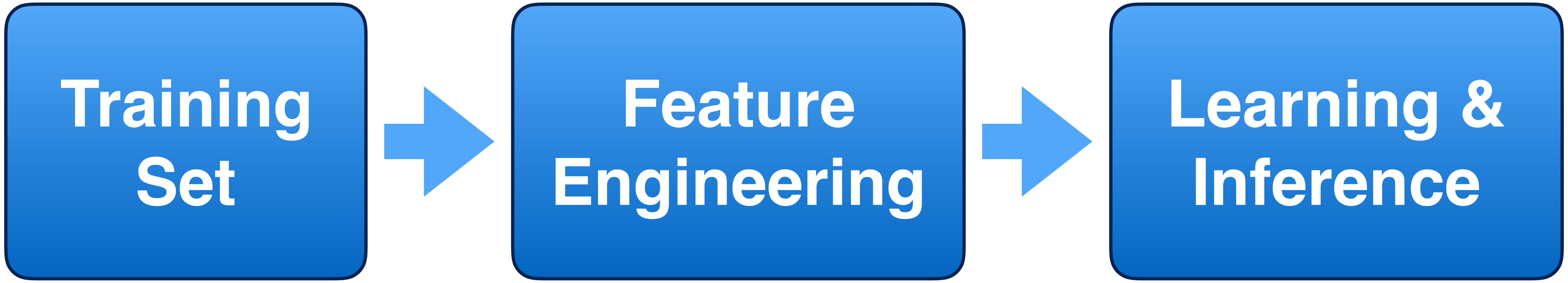
Extract Spouse Mentions from Text

TASK: Build a **knowledge base** of married couples by extracting mentions of **spouses** from news articles

`Jeffrey Navin, 56, and his wife, Jeanette, 55, a school paraprofes`
`on Facebook by Rachel Hattingh and her husband Graham Marshall, a`
`Brecht-Schall was married to actor Ekkehard Schall, a stalwart of`

Sentences containing mentions of married couples

Traditional Machine Learning Approach…

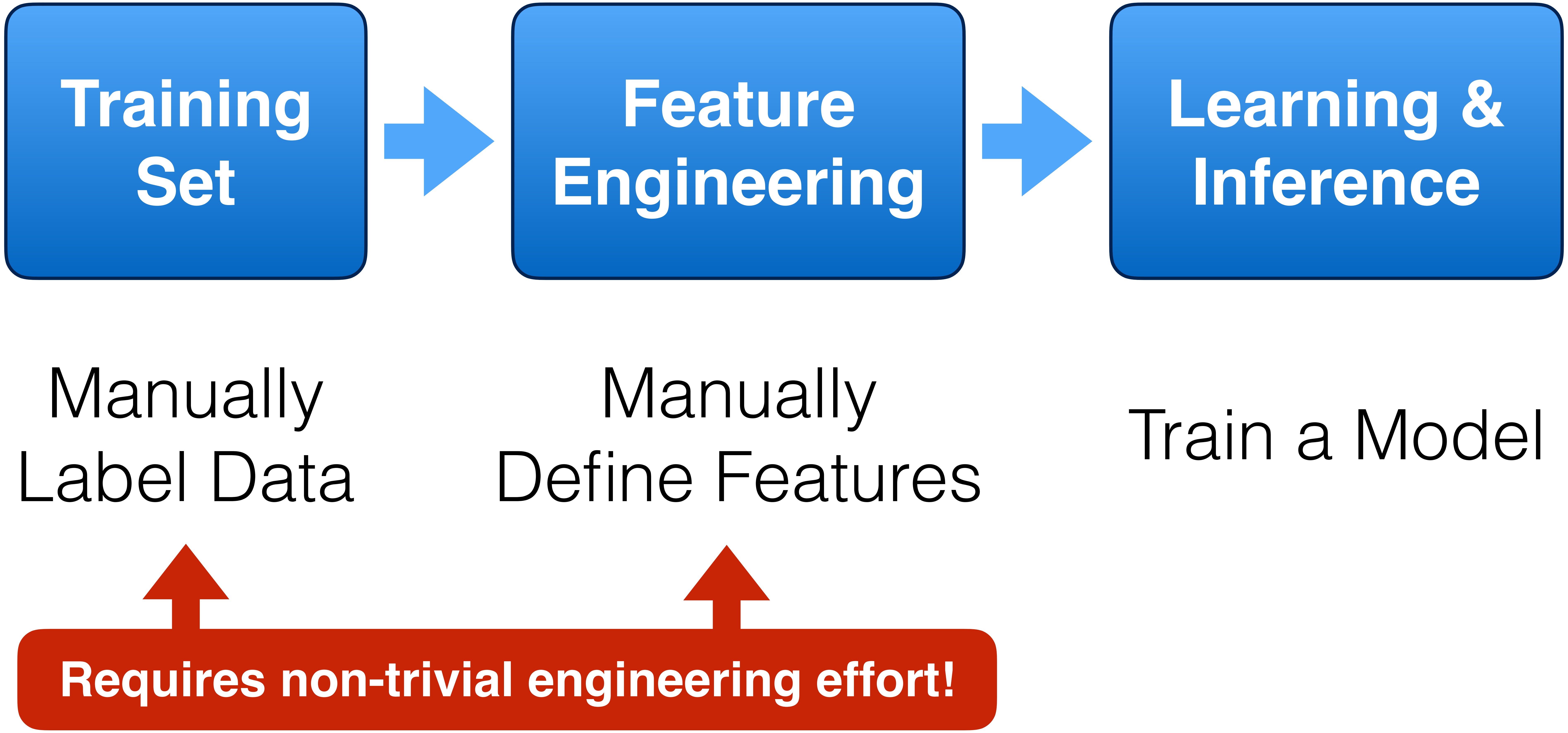


Manually Label Data

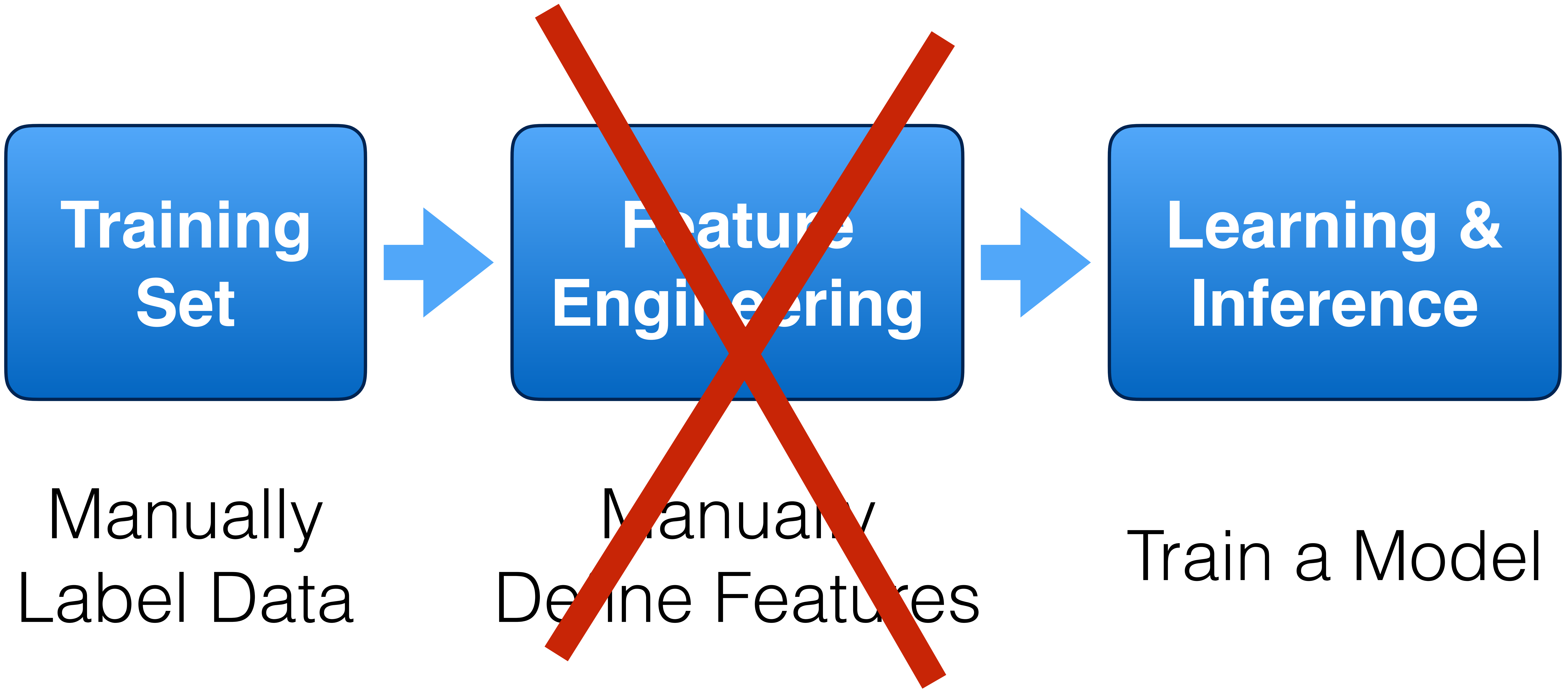
Manually Define Features

Train a Model

Traditional Machine Learning Approach…



Traditional Machine Learning Approach…



Deep Learning Killed Feature Engineering

Traditional Machine Learning Approach…

but we still need to label a bunch of data!

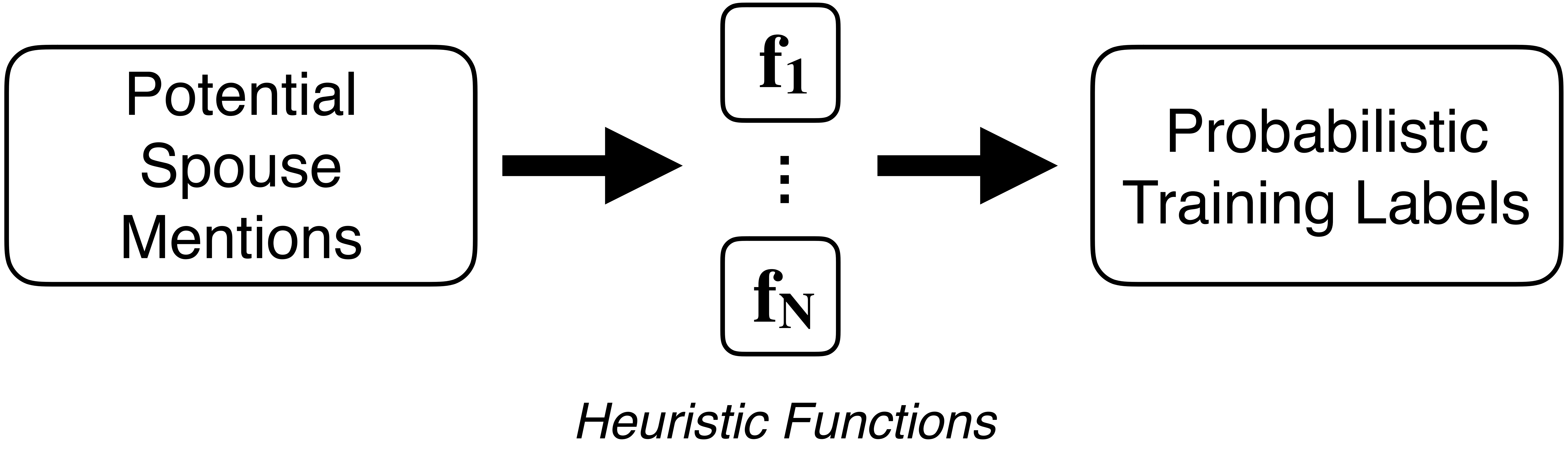
Ellen DeGeneres and wife Portia De Rossi have seemingly shut down divorce rumors with a joint outing in Los Angeles. ✓

Khloe Kardashian says she's DEFINITELY down to marry Tristan Thompson ... even though he hasn't exactly proposed yet. ✗

Repeat hundreds or thousands of times…

Snorkel / Data Programming Approach…

Write *heuristics* to noisily label data!



Programmatically generate training data

Labeling Functions: Intuition and Overview

Labeling Functions

```
Side-by-Side was started on Facebook
by Rachel Hattingh and her husband
Graham Marshall, a London homeless
charity chief executive, from
Stanford-le-Hope.
```

Is this a true spouse mention?
What evidence informs your decision?

Labeling Functions

```
Former U.S. president **Barack Obama** and
first lady **Michelle Obama** arrive to
talk about the Obama Presidential
Center during a community event at the
South Shore Cultural Center on May 3
in Chicago, Illinois.
```

Is this a true spouse mention?
What evidence informs your decision?

Labeling Functions

Human annotators leverage
real-world knowledge, **context**,
and **common-sense heuristics**
to make labeling decisions

We can model parts of this process by
encoding these rules as functions …

Labeling Functions

Labeling Functions (LFs)

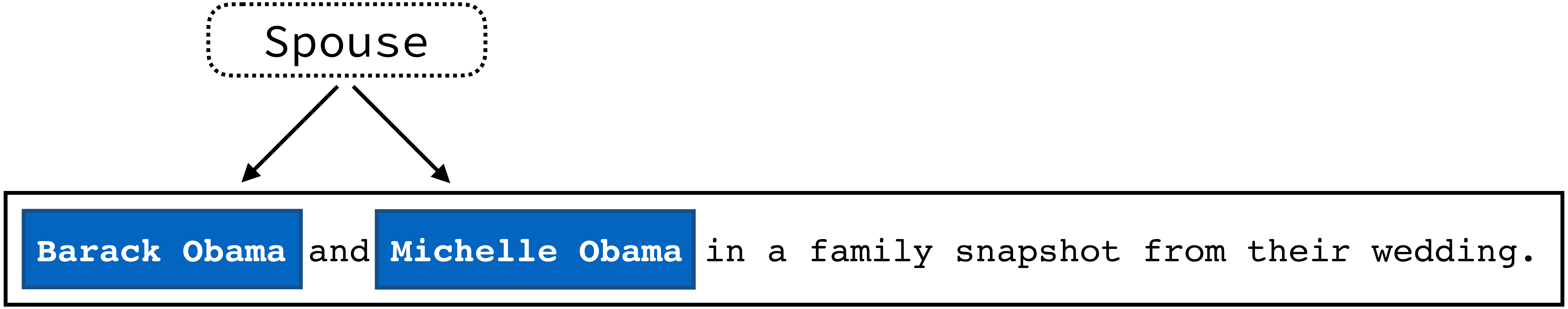
Black box functions that label subsets of data

`{-1, 0, 1}`

`{Negative, Abstain, Positive}`

Candidates

All pairs of **people's names** in a sentence



Labeling Functions

Candidates includes **true** and **false** instances

SENT_ID 1: `Jeffrey Navin, 56, and his wife, Jeanette, 55, a scho`

SENT_ID 2: `Khloe Kardashian says she's DEFINITELY down to marry`

`(Jeffrey Navin, Jeanette)` ✓

`(Khloe Kardashian, Tristan Thompson)` ✗

Labeling Functions

Goal: Provide (potentially weak) correlated signal with true class labels

Apply labeling functions to all candidates

Predict both **positive** and **negative** labels

Labeling Functions

INSIGHT

People with the same last name *might* be married

… photos taken of President **Barack Obama** and first lady **Michelle Obama** during …

INSIGHT

If '**boyfriend**' or '**girlfriend**' appear between people mentions, the pair are probably *not* married

… **Pippa** is engaged to her hedge fund manager **boyfriend James Matthews** …

Labeling Functions

Implement these rules as Python functions

```
def LF_same_last_name(c):
    """
    Label as positive if both
    """
    p1_last_name = last_name(c.person1.get_span())
    p2_last_name = last_name(c.person2.get_span())
    if p1_last_name and p2_last_name and p1_last_name == p2_last_name:
        if c.person1.get_span() != c.person2.get_span():
            return 1
    return 0
```

```
def LF_dating(c):
    dating = {'boyfriend', ‘girlfriend'}
    return -1 if len(dating.intersection(get_between_tokens(c))) > 0 else 0
```

Labeling Functions

Labeling functions can be **noisy**

People with the same last name *might* be married

TRUE	PREDICTED	
✓	✓	… photos taken of President **Barack** **Obama** and first lady **Michelle** **Obama** during …
✗	✓	**Mary-Kate** **Olsen** and **Ashley** **Olsen** (born June 13, 1986), also known as the Olsen twins collectively…
✓	—	**Tom Hanks** reveals his 28-year marriage to **Rita Wilson** almost never happened.

Labeling Functions:
Design Strategies

Labeling Functions

Jeffrey Navin, 56, and his wife, Jeanette, 55, a school parapr

ook by Rachel Hattingh and her husband Graham Marshall, a London h

Ellen DeGeneres and wife Portia De Rossi have seemingly

Previously, we used common-sense **patterns** or **keywords** to label a person pair as married or not

Labeling Functions

Jeffrey Navin, 56, and his wife, Jeanette, 55, a school parapr

ook by Rachel Hattingh and her husband Graham Marshall, a London h

Ellen DeGeneres and wife Portia De Rossi have seemingly

Pattern-based Labeling Functions

Labeling Functions

INSIGHT

If ‘boyfriend' or ‘girlfriend' appear between people mentions, the pair are probably *not* married

… **Pippa** is engaged to her hedge fund manager **boyfriend** **James Matthews** …

These are implemented using **string matching** via **regular expressions** and other heuristics

Labeling Functions

We can also use other sources of information to generate LFs

Distant Supervision Labeling Functions

These use an existing database of known facts to generate noisy labels

Labeling Functions: Distant Supervision

```
def known_spouse(x):
  pair = (x.person1_id, x.person2_id)
  return 1 if pair in KB else 0
```

Former U.S. president **Barack Obama** and first lady **Michelle Obama** arrive to talk …

Knowledge Base (KB)

CONTAINS (A B)

Label = True

Labeling Functions: Distant Supervision

UMLS

Unified Medical
Language System

Many **public knowledge bases**
are available, especially in **biomedicine**

Labeling Functions: Distant Supervision

Public semantic **knowledge base,** let's use this resource for distant supervision

http://wiki.dbpedia.org/

Labeling Functions: Scoring Metrics

Labeling Function: Metrics

How do we assess the quality
of our labeling functions?

Labeling Function: Metrics

Accuracy: The percentage of candidates a labeling function labels correctly

Coverage: The percentage of all candidates that are labeled by >= 1 LFs

Conflict: The percentage of candidates with >1 labels that disagree

Labeling Function: Metrics

Assessing empirical accuracy requires some ground truth labels

Dev Set: A small set (~100 candidates) of human labeled examples we can use to guide LF development

Labeling Function: Metrics

Ideally, we want **high-coverage**, **high-accuracy** LFs

LFs need to label with **probability better than random chance**

Conflict is actually good — it allows our algorithm to learn information about the LF

Terminology

$$\text{Precision} = \frac{tp}{tp + fp}$$

How often a predicted label is correct

$$\text{Recall} = \frac{tp}{tp + fn}$$

Given the known total number of positive instances, how many were labeled correctly

$$\text{F}_1\text{-score} = \frac{precision * recall}{precision + recall}$$

Harmonic mean of precision and recall

Snorkel API
(Hands-on Exercises)

Snorkel API

Open Tutorial Notebook

Workshop_1_Snorkel_API.ipynb

- Introduce Jupyter notebooks
- Fill out your email/username
- Introduce Candidate classes
- Complete exercises 1 & 2

Writing Labeling Functions (Hands-on Exercises)

TIME: 60 Minutes

Writing Labeling Functions

Open Tutorial Notebook

Workshop_2_Writing_Labeling_Functions.ipynb

- Introduce labeling function factories
- Complete tutorial examples

Generative Model:
Unifying Supervision

Terminology

Generative Model

$$P(x,y)$$

Learn the joint distribution of (x,y)

Example Classifiers

Naive Bayes

Discriminative Model

$$P(y|x)$$

Learn the conditional probability of y given x

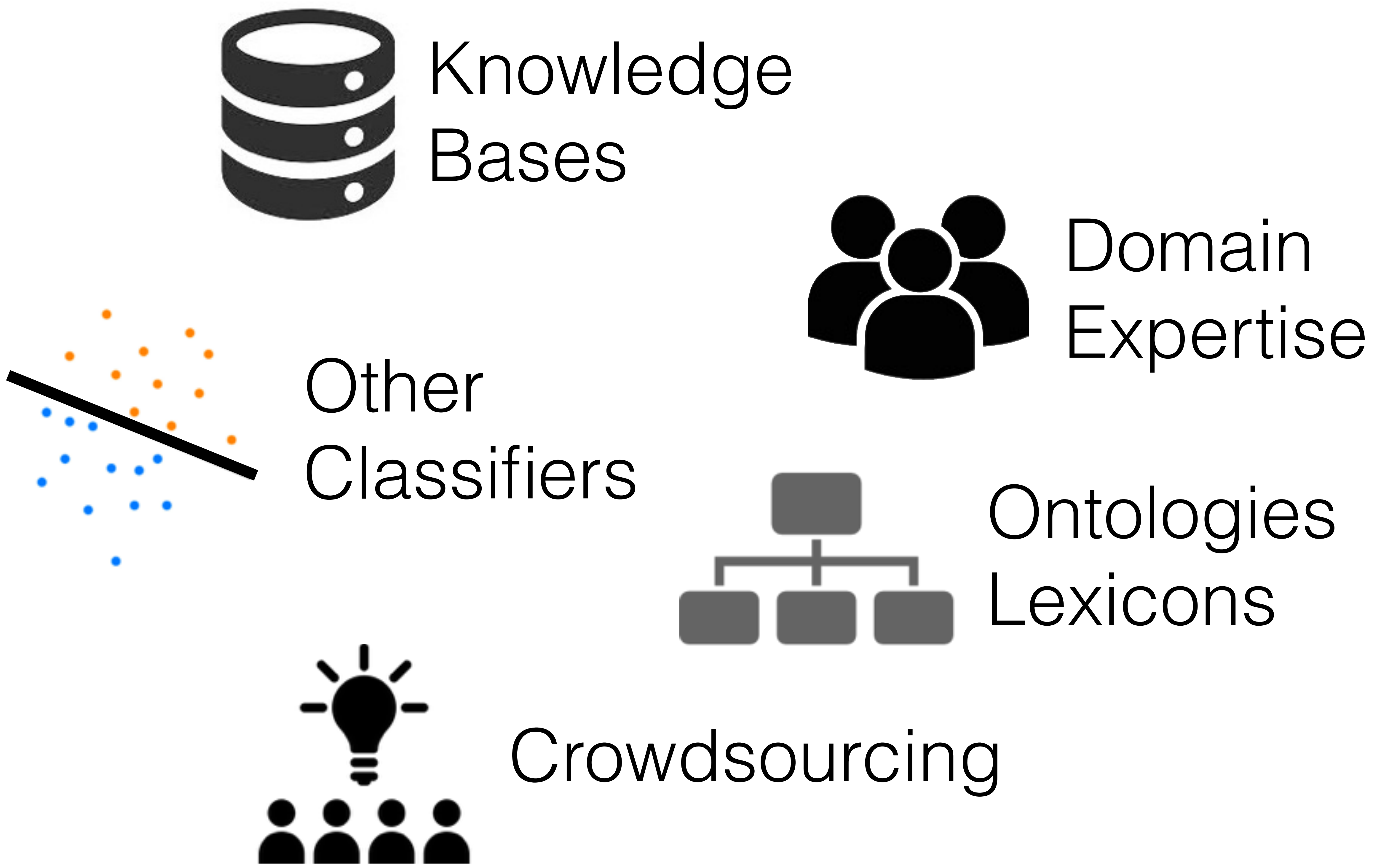
Example Classifiers

Support Vector Machine (SVM)
Logistic Regression
Deep Neural Networks (LSTMs)

Generative Model: Unifying Weak Supervision

Labeling functions allow for
radically weaker labels

These labels can be noisy, conflicting, and come from a **variety of inputs**



Key Idea: Labeling functions encode all these forms

Intuition: How Does it Work?

Simplest way to unify LFs is
unweighted majority vote

Intuition: How Does it Work?

As long as most people vote correctly ($p > 0.5$), adding more people improves the accuracy of majority vote*****

*** Condorcet's Jury Theorem**

Intuition: How Does it Work?

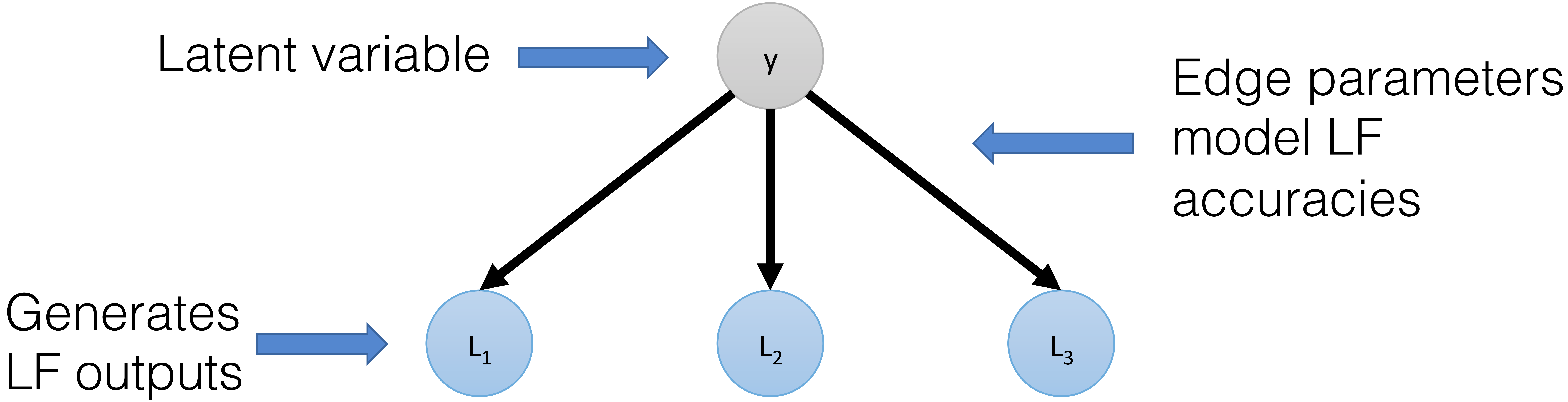
LFs have different **latent accuracies**

Unweighted majority vote ignores this!

Intuition: How Does it Work?

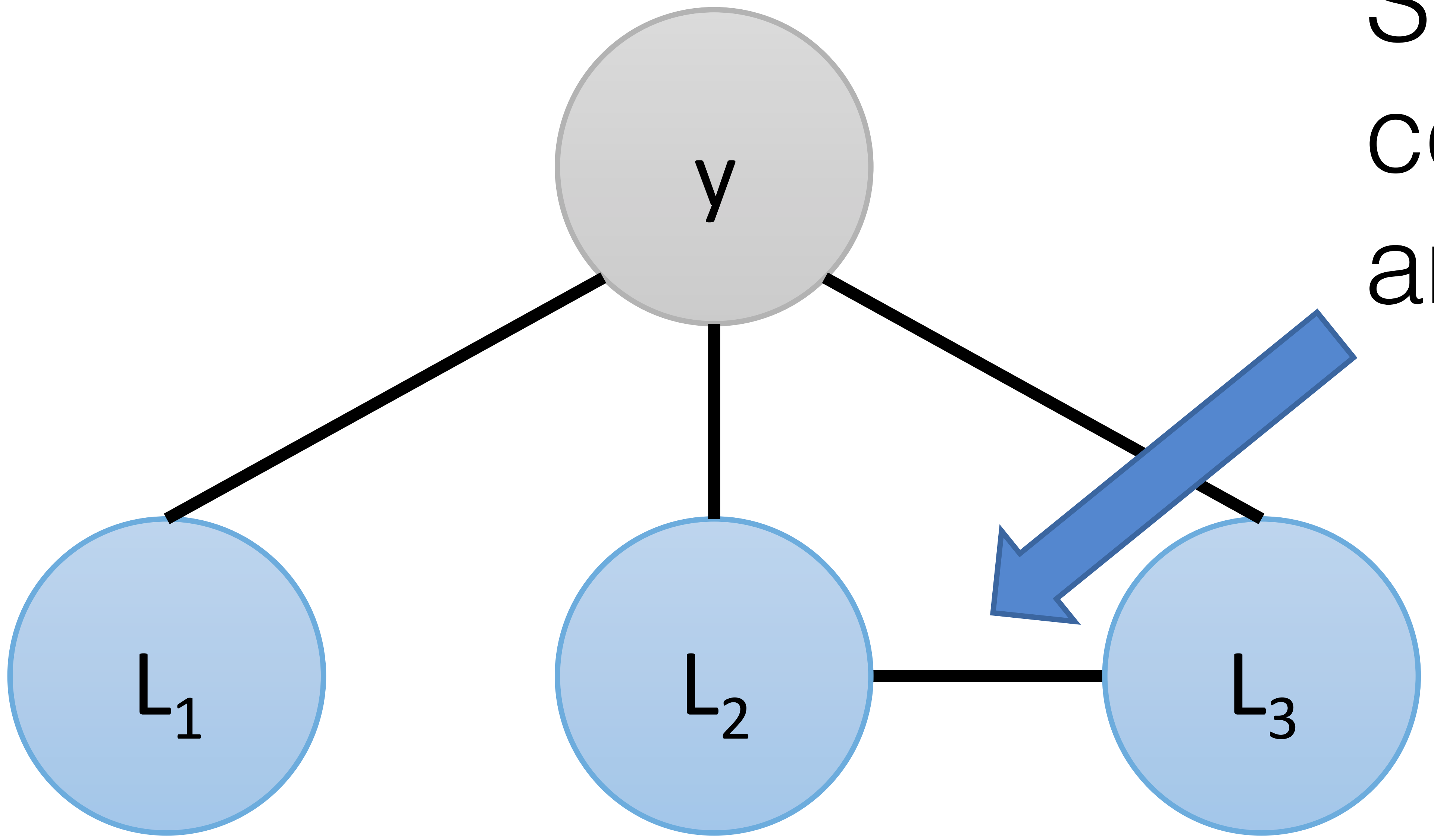
We want to learn these latent accuracies **without labeled data** by leveraging **overlap** and **conflict** of LFs

Generative Model: Unifying Weak Sources …



We maximize the marginal likelihood of the LFs to learn parameters
Intuitively, compares their agreements and disagreements

Generative Model: Structure Learning



[Bach et al., ICML 2017]

Snorkel can automatically detect correlations and other dependencies among LFs to correct their accuracies

Adds, on average, a **1.5 F1 boost** to models — for free

See the Snorkel blog post for more details

https://hazyresearch.github.io/snorkel/blog/structure_learning.html

Structure Learning

Data programming assumes LFs make **independent labeling decisions**

If LFs make **correlated decisions**, independent of the true label, the MLE of the parameters will **overweight LFs latent accuracies**

Structure Learning

When does this happen?

- Using **multiple**, **overlapping ontologies** for distant supervision
- LFs only differ due to **tunable parameters**, like context window size.
- Many more!

Training the Generative Model (Hands-on Exercises)

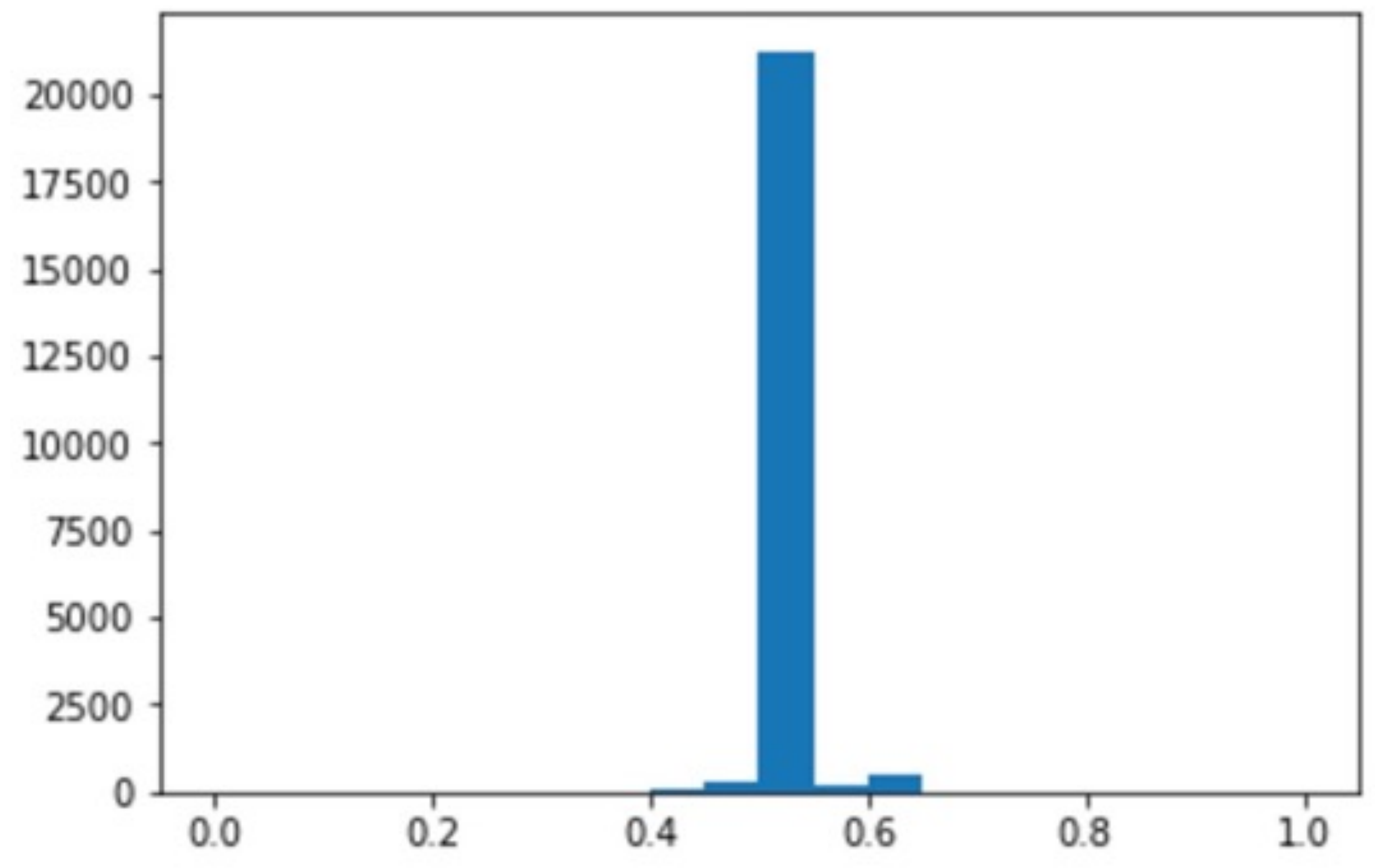
Training Generative Models

Open Tutorial Notebook

Workshop_3_Generative_Model_Training.ipynb

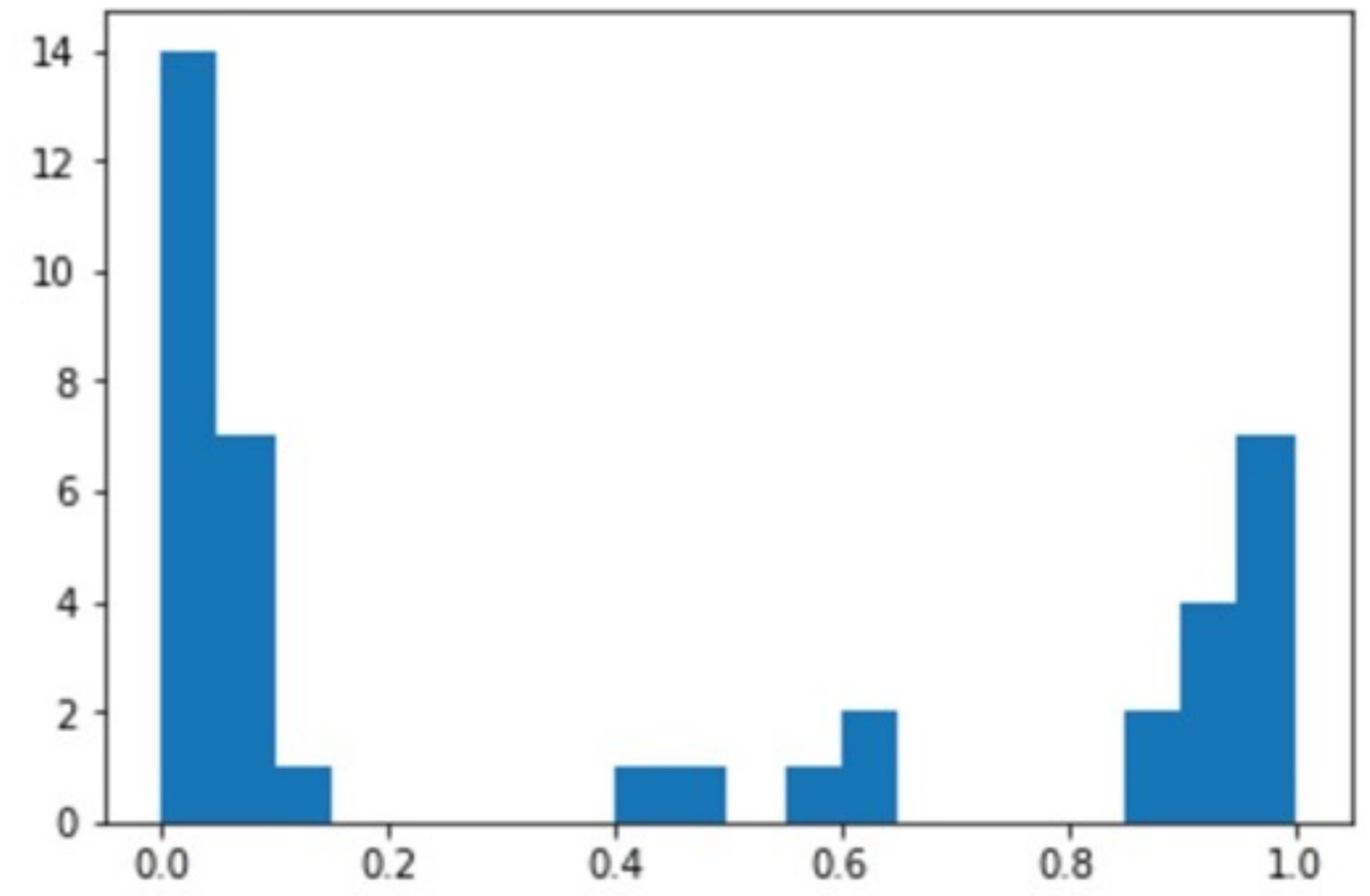
- Majority vote
- Training the generative model
- Interpreting Marginals
- Learning dependencies

Generative Model: Interpreting Marginal Distributions



This is probably the first set of marginals you'll generate. These are **BAD!**

Everything's clustered at 0.5, i.e, **no labels**



This are the marginals you want!. These are **GOOD**.

Clear differentiation between 0.0 / 1.0

TAKE 2

Refine Writing Labeling Functions
(Hands-on Exercises)

TIME: 45 Minutes

Discriminative Model:
“Compiling” Rules into Features

Snorkel API

Open Tutorial Notebook
Workshop_4_Discriminative_Model_Training.ipynb

- Train an LSTM model

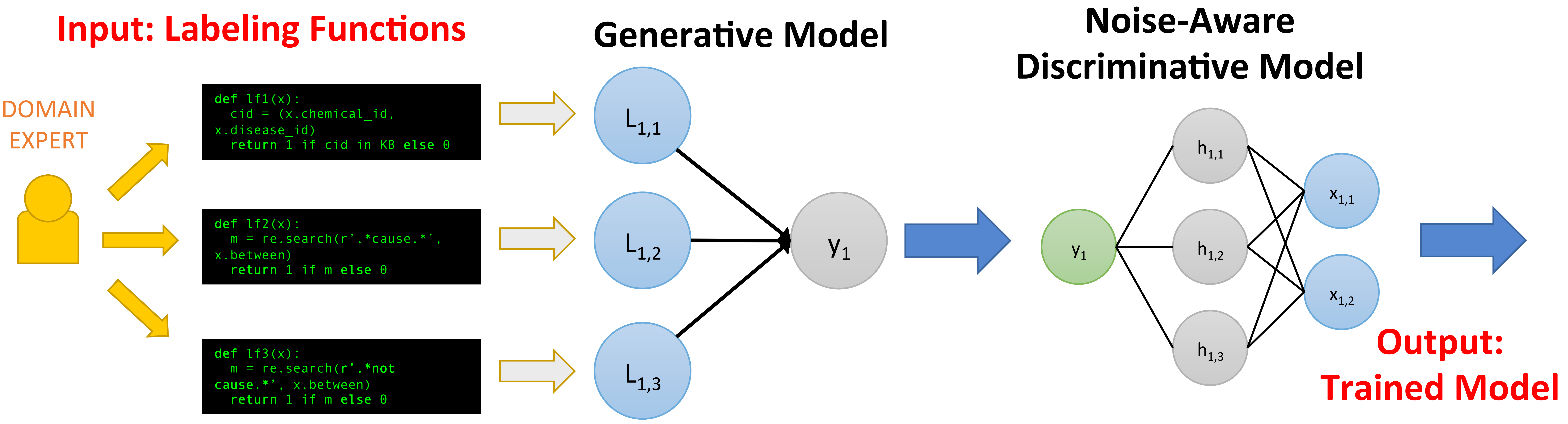
This takes ~10 minutes. Start now!!

Discriminative Model

The output of the generative model is a set of **probabilistic training labels**

We now want to use these labels to train our final discriminative model

Discriminative Model: Full Snorkel Pipeline



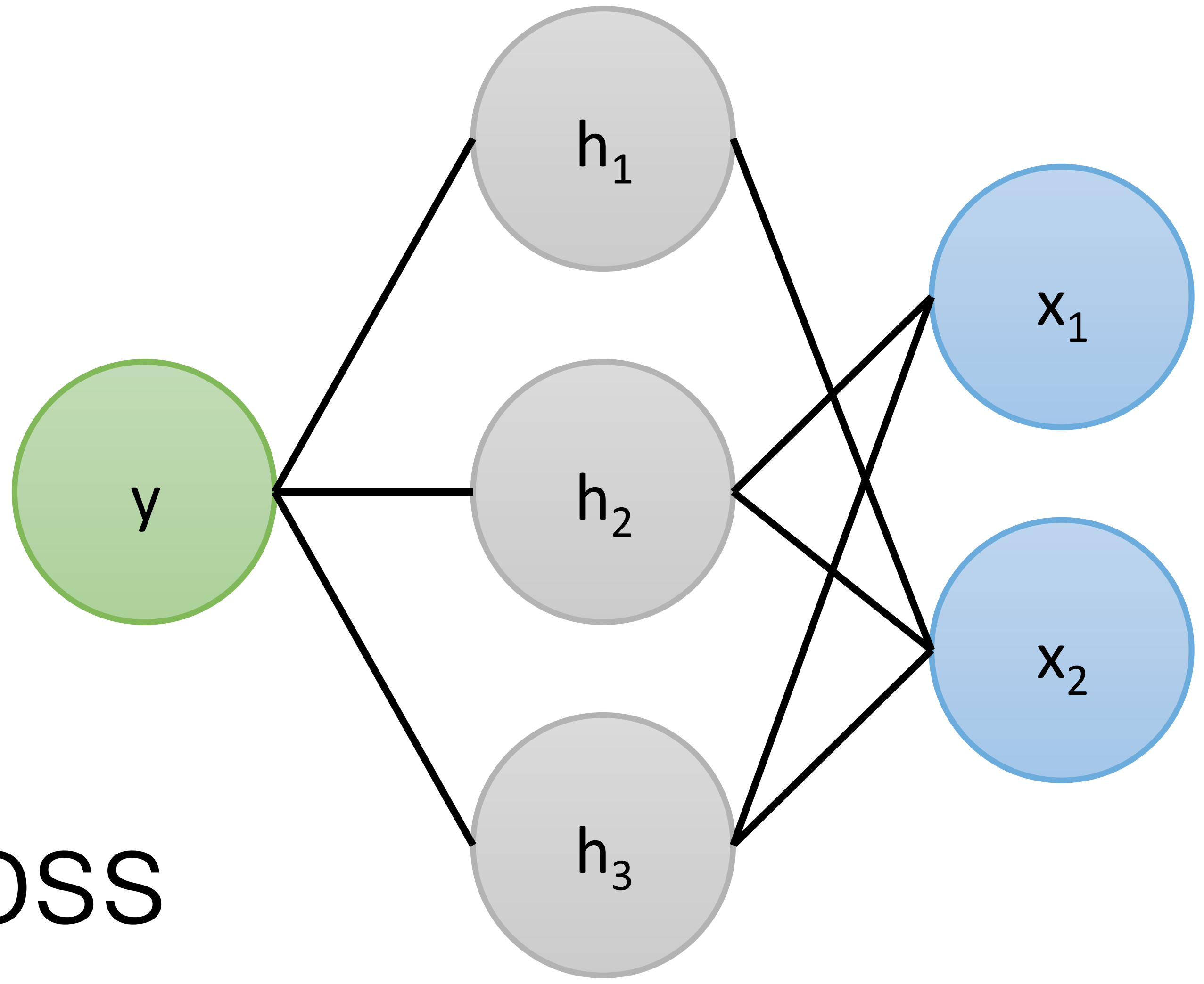
Users write scripts to label training data

We model this process to denoise it

We use this to train e.g. a deep learning model!

Discriminative Model

Train on marginals from generative model



Minimize noise-aware loss

Generalization error decreases at same asymptotic rate as in supervised setting, except **in amount of unlabeled data**

[Ratner et al., NIPS 2016]

Training a *Noise-aware* Discriminative Model

Supervised Learning Loss Function

$$\hat{w} = \arg\min_w \frac{1}{N}\sum_{i=1}^{N} l(w, x^{(i)}, y^{(i)})$$

Noise-aware loss

$$\hat{w} = \arg\min_w \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{(y,\Lambda)\sim\pi}[l(w, x^{(i)}, y^{(i)} = y)]$$

Simple change for Logistic Regression, SVMs, LSTM (neural networks)

Discriminative Model

Why can’t we just use the generative model for our final predictions?

The discriminative model learns a **feature representation** of our **LFs**

This makes it better able to generalize to unseen candidates

Discriminative Model

As a result, we see much better recall!

	Precision	Recall	F1
Majority Vote	76.4	67.3	71.5
Generative Model	67.4	**77.9**	72.3
CRF	**81.5**	75.8	78.5
BiLSTM-CRF	80.7	77.6	**79.1**

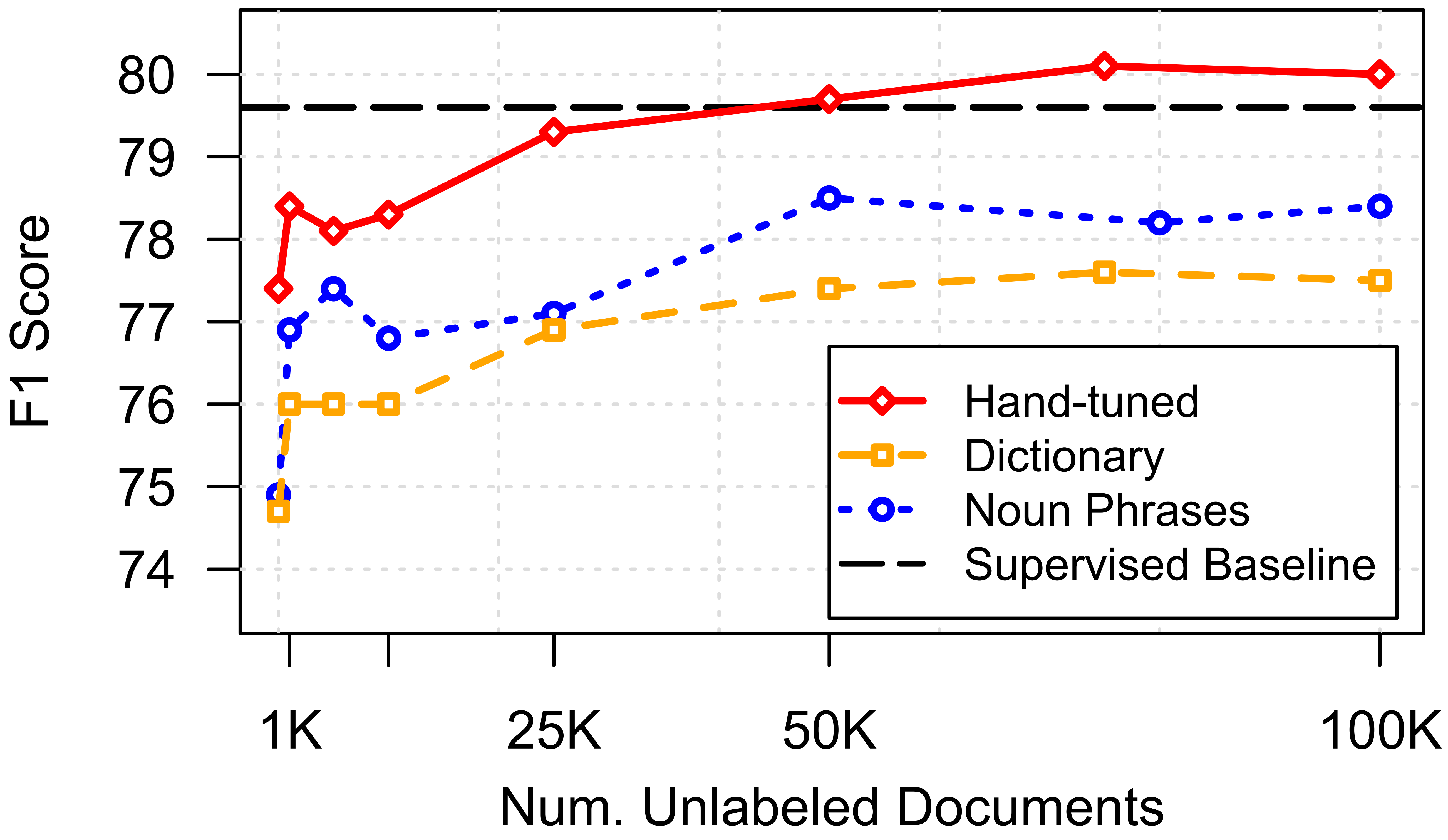
— CDR disease name tagging

[Fries et al., 2017]

Discriminative Model

We can now **automatically generate large-scale training sets**

We can **match or exceed** supervised learning performance



Tagging disease names in PubMed

[Fries et al., 2017]

TAKE 3

Refine Writing Labeling Functions (Hands-on Exercises)

TIME: 45 Minutes

Application Development:
Introducing Schemas and Evaluation Plans

Application Design

Two critical questions for any new application

What **information** am I **extracting**?

Once extracted, what is the **utility**
of this **new information**?

Application Design: Project Template

We've provided a project template for tomorrow's discussion

1. Motivation
2. Task Overview
3. Data Set Overview
4. **Schema Design**
5. Validating Your Extraction Models
6. External Utility

Application Design: Schema

Schema: The formal definition of what we are extracting from text. This is the structured representation of our facts.

```
Spouse ( PERSON, PERSON )
Chemical-induced Disease (CHEMICAL, DISEASE)
Side Effect ( DRUG, SYMPTOM/SIGN )
```

Formally defining these entities and relations is the most important step in building a Snorkel application!

Contact Us

Code Issues?

GitHub: Snorkel Issues

http://snorkel.stanford.edu
https://github.com/HazyResearch/snorkel

jason-fries@stanford.edu
ajratner@cs.stanford.edu
bach@cs.stanford.edu
joyku@stanford.edu