Programming is Changing & Software 2.0

Chris Ré

Stanford

My (current) plan

Section 1: *Overview, Motivation, and Applications.*

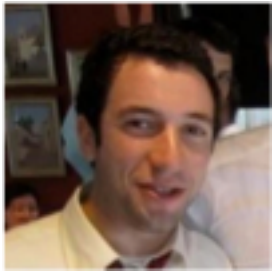
- The **why** we're doing it. You are great researchers, I want to infect you with this direction. It changed our whole lab's direction!

- **Activity** 1: Your 1st Software 2.0 app? Weak supervision in Snorkel.

Lecture 2: *Algorithms & Theory*. *This is what we actually do all day.*

- **Activity** 2: Play with the generative model, try your hand at image processing, burn credits!

Sessions 2 & 3.5 onward!



Alex Ratner (Market Next Year)

Jason Fries, PhD

They are in Palo Alto, and they will run the session—they do the real work!

Context: Why am I telling you this?

- I want to share with you how we view research and the personal story of how it came together.

- You are the next generation.
  - You can write papers.
  - You can do research.
  - Will you try to change the world? Or try to get tenure/job?

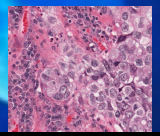
I did not appreciate until much later how important why is.

How we got here.

Dark Data

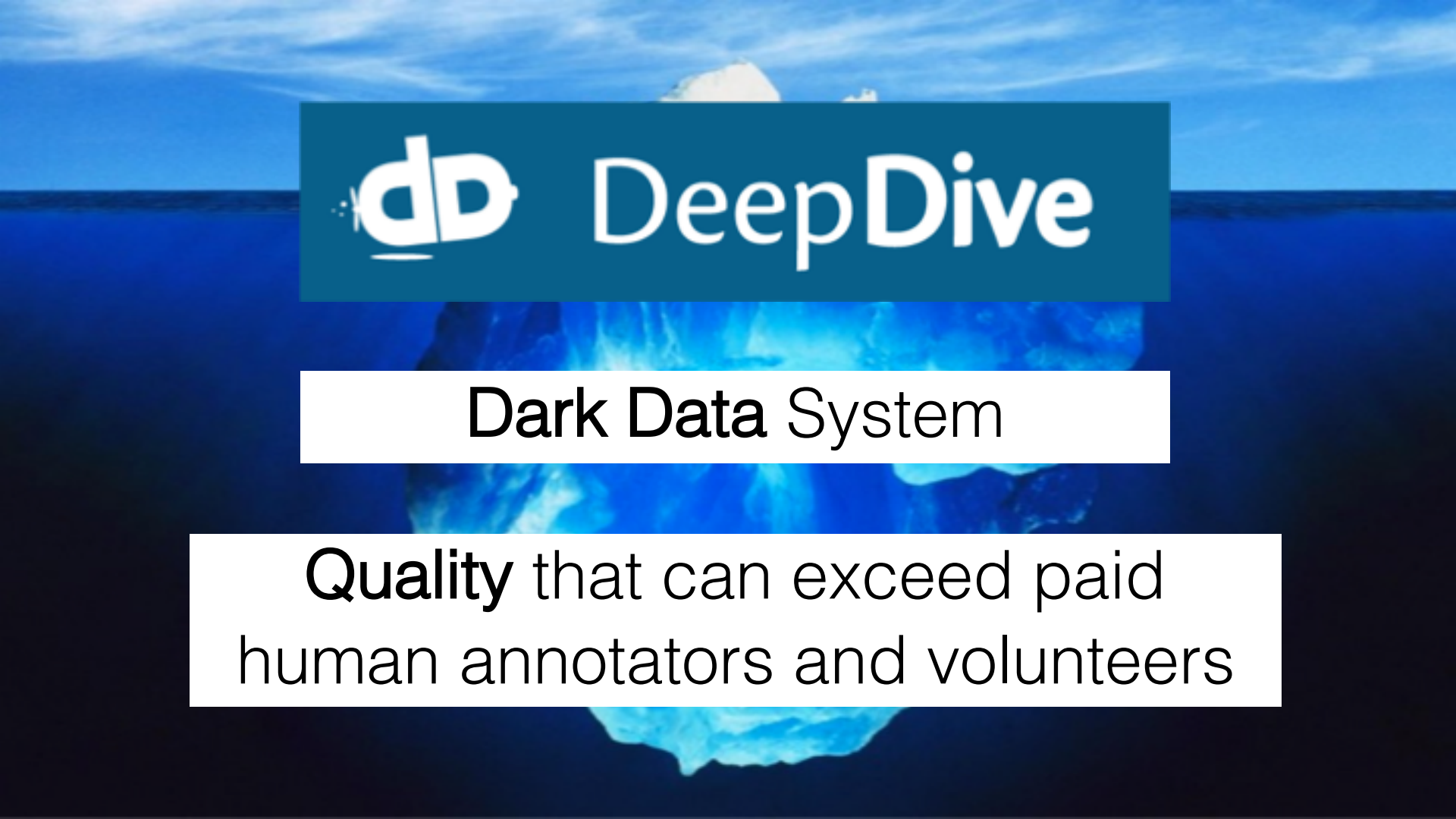
Data easy to process by machines

Scientific articles &
government reports.



Medical Images

Dark Data.
Valuable & hard to process.


DeepDive
Dark Data System
Quality that can exceed paid human annotators and volunteers

Bringing dark data to light may help
improve science, **business**, and **society**.

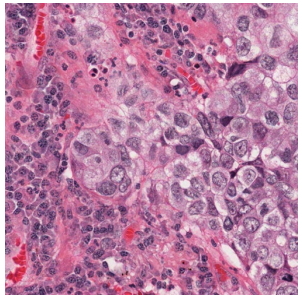
Ce Zhang
(ETH)

Feng Niu
Lattice cofounder
Apple

Dark data can improve science, business, and society.

Biodiversity

Drug Repurposing



Lung Cancer Prognosis

Fight human trafficking

Chris White

LATTICE

Lattice now part of Apple

Human Trafficking on the Web…

Juliana Freire NYU

DARPA MEMEX

Hypothesis: Trafficked individuals offer ***lower cost*** and ***riskier*** sexual services.

In Plain sight: Web ads for sexual services

Challenges:
1. Need **high-resolution information** to build model.
2. Scientific papers are **clear**—*dark web is* **obfuscated**.

Mike Cafarella (Michigan)

Dark Data helps with Societal Problems

100M ads read with human-caliber quality.

- Child predators human traffickers arrested in multiple jurisdictions across the US.

2016 Presidential Award for Extraordinary Efforts to Combat Trafficking in Persons. Chris White.

the WHITE HOUSE PRESIDENT BARACK OBAMA

DARPA MEMEX

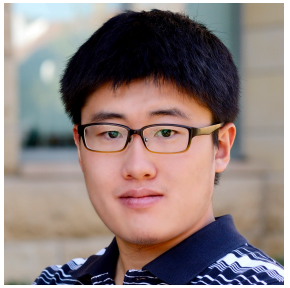
LATTICE

Fighting Human Trafficking with Dark Data

Jared Dunnmon

Mike Cafarella

Saeideh Shahrokh Esfahani



Sen Wu

Reliable, scalable **KBC from semi-structured sources** like raw HTML allows us to rapidly ingest the amount of data required to train **ML models. Continuing even today!**

accenture

https://github.com/HazyResearch/fonduer

Some Lessons.

- This is a traditionally a bad idea for a career.
  - Almost none of this came out in traditional papers.
  - There were several times I was told explicitly not to do this work.

- I couldn't have been happier about what I learned.
  - You pay a lot of money to do this job. Drive it like you stole it.

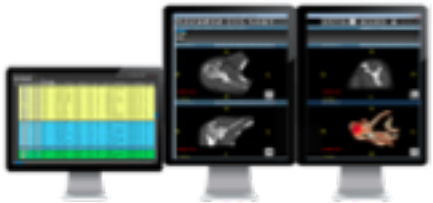
- You can always build something better with a team.
  - I owe several people my career: Mike Cafarella (grad school friend), Feng Niu and Ce Zhang (1st and 2nd students), and Chris White.
  - I owe my wife too, but that's a given.

Commodity extractors in the hands of the right people can do good.

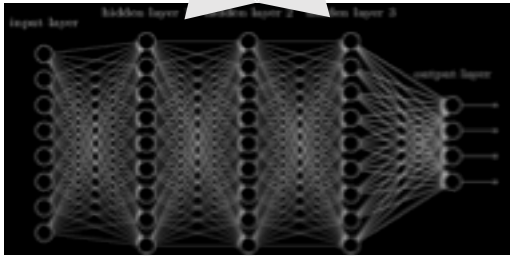
Can we radically deskill this process?

Interlude: … not just deskill …
Silicon Valley Reality Distortion Bubble

Software 2.0 is eating Software 1.0



1000x Productivity: Google shrinks language translation code from 500k LoC to 500.

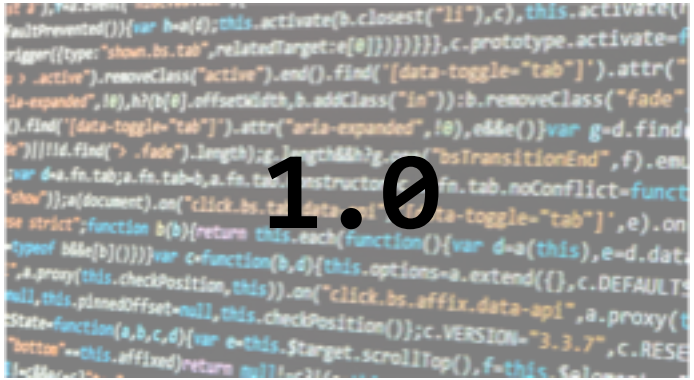


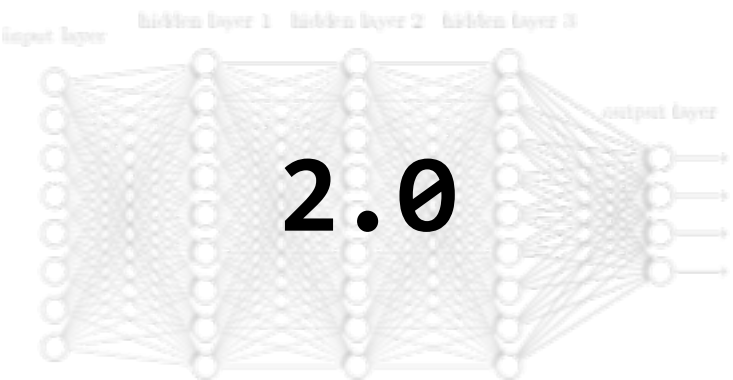
Classical problems (ETL/cleaning/tuning software/network) are moving to ML first Holoclean.io (UW), Peloton (CMU), Pensieve (MIT) .

https://jack-clark.net/2017/10/09/import-ai-63-google-shrinks-language-translation-code-from-500000-to-500-lines-with-ai-only-25-of-surveyed-people-believe-automationbetter-jobs

"Software 2.0", Andrej Karpathy, https://medium.com/@karpathy/software-2-0-a64152b37c35

Training data: the new input to the 2.0 stack





- Input: Code
- Compiled to: Machine instructions

- Input: Training data
- Compiled to: Learned parameters

https://medium.com/@karpathy/

ONNX

Why? Easier to build and deploy

Build products faster. Speed is amazing.

Deploy is critical: NNs “new JVM”

- Regular run-times. (Blas Calls/No Allocation).
  - Qualification easier == ship faster.
- In a parallel life, work on software-hardware for ML
  - See Dawn.cs.stanford.edu for more
- *Ask me about systems and optimization work!*

SambaNova SYSTEMS

Kunle Olukotun

Our bet two years ago…

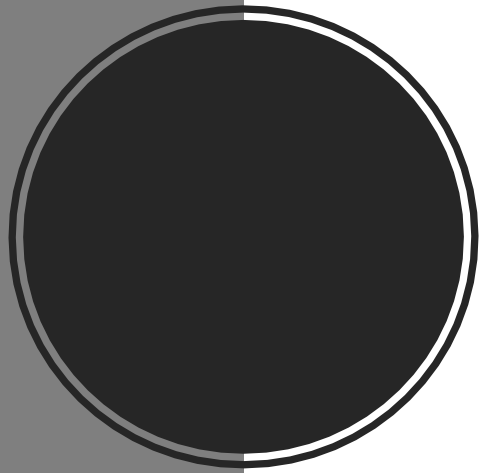
- Increasing ubiquity of 2.0 via commodification of deep learning.
  - We called it **Data Programming** [NIPS16]
- *Engineers spend their time shaping training data:* Why isn't their any mathematical or systems structure?

KEY IDEA:

Training data is the critical interface to *program* Software 2.0…

Goal *leverage varied quality sources of training data via higher level abstractions.*

Silicon Valley Reality Distortion Bubble end?



snorkel

Radically easier to use ML systems

ML is **hard** but should be **easier**

- Simple classifiers and regression
- Entity and relationship extraction
- Entity linking and normalization

Stretch goal: world-class quality in hours.

Snorkel @ Snorkel.Stanford.edu

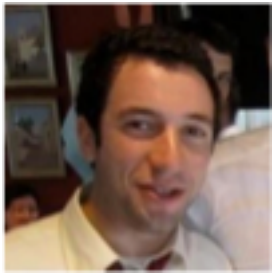
snorkel

The Real Work

Stephen Bach (Bach)

Chris De Sa (Cornell)

Henry Ehrenberg (Facebook)



Alex Ratner (Market Next Year)

Paroma Varma

Three takeaways

Ease we can deskill many ML tasks & Change what experts do.

Interesting This perspective raises new twists on classic questions.

Much more to do. Data augmentation & blog of related work!

Why now and an overview…

Tech Push:
The Rise of Automatic Feature Libraries

Deep learning is becoming a **commodity**

- **Success or failure of an application seems may depend less on the model—more what data its fed.**

Automatic feature libraries need **large training** sets.

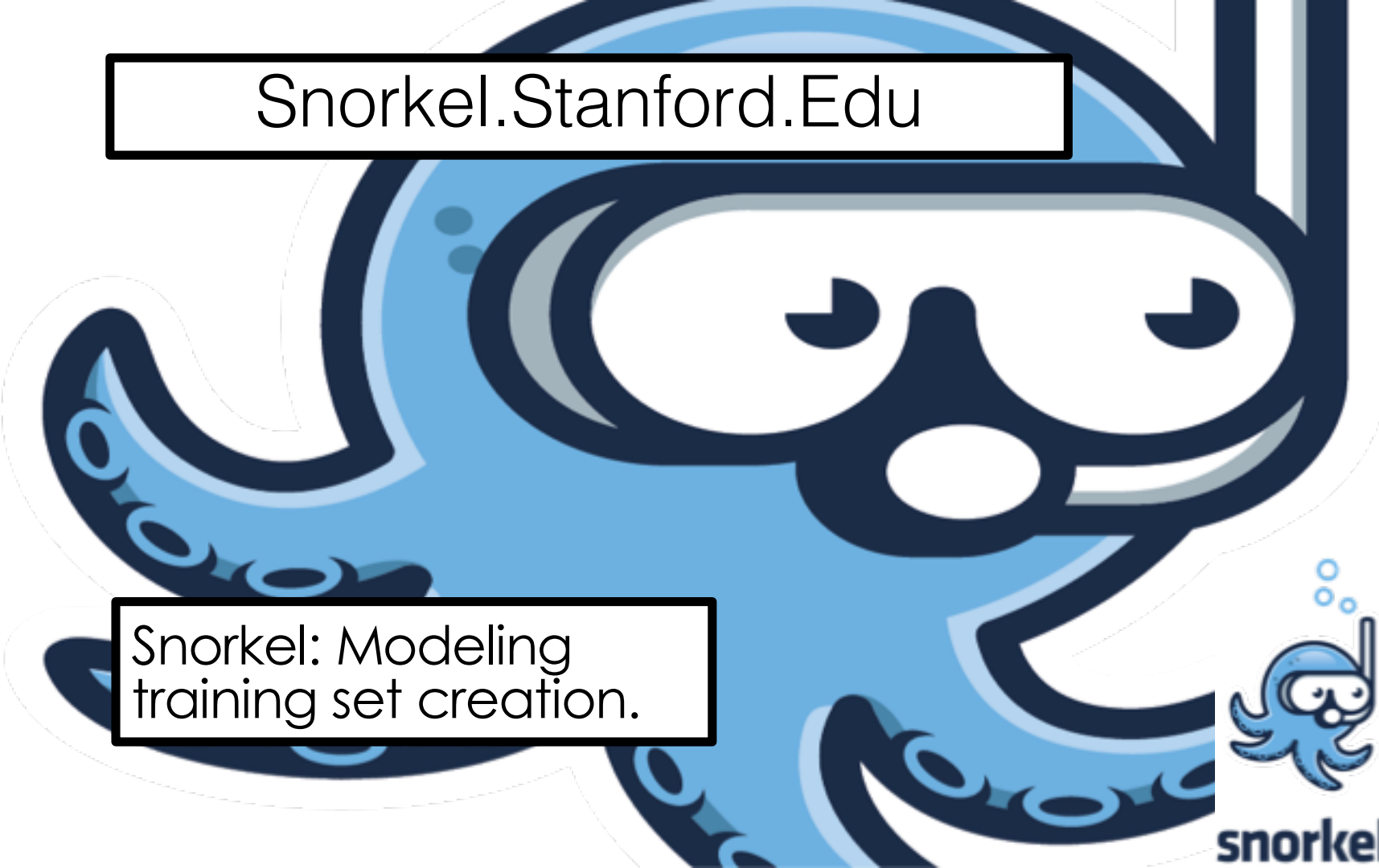
Creating **training sets** is often the bottleneck.

The *New* New Oil

A Fundamental Problem in Machine Learning

Key idea: Model **process** or provenance of training set creation.

Snorkel.Stanford.Edu



Snorkel: Modeling training set creation.

snorkel

Data Programming Pipeline in Snorkel

Input: Labeling Functions, *Unlabeled data*

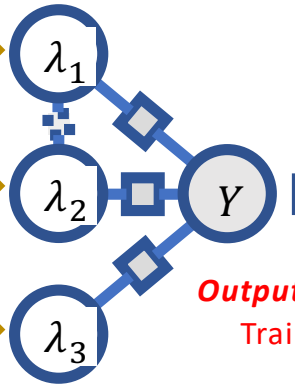
DOMAIN EXPERT

```
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
```

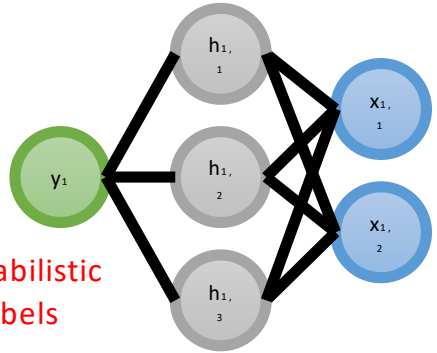
```
def lf3(x):
    m = re.search(r'.*not
cause.*', x.between)
    return 1 if m else 0
```

Generative Model

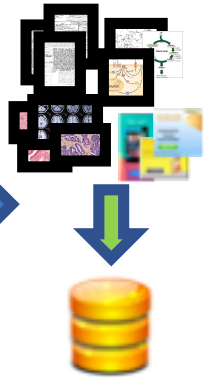


Output: Probabilistic Training Labels

Noise-Aware Discriminative Model



Ex. Application: Knowledge Base Creation (KBC)



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

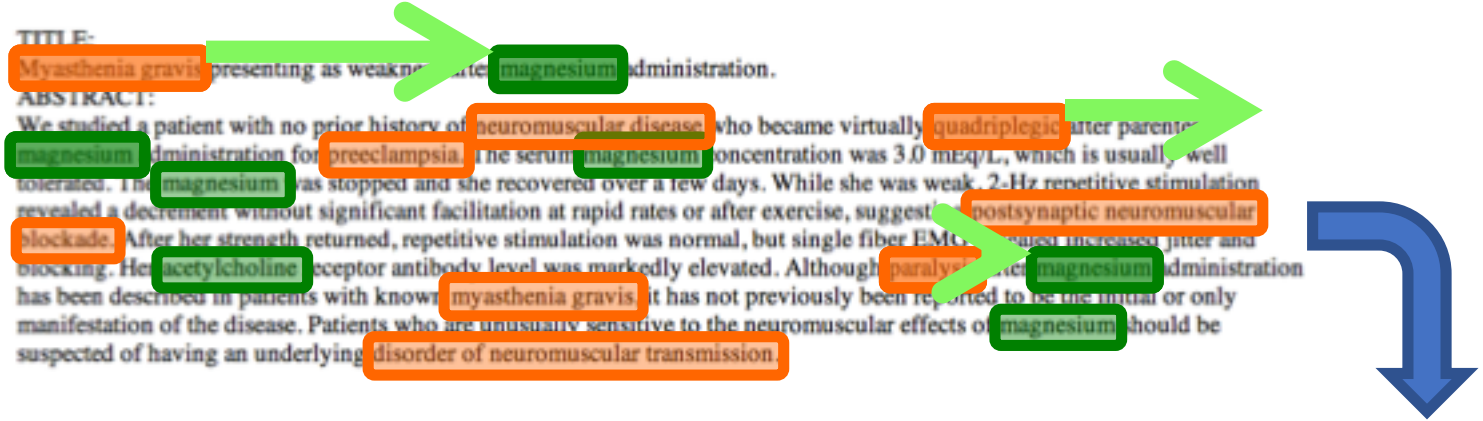
Case Study: **Lightweight** Extraction

- Dark data extraction systems still take **months or years** to build using state-of-the-art ML systems

- We want to build systems that answer questions in ***hours to days***

What is holding us back?

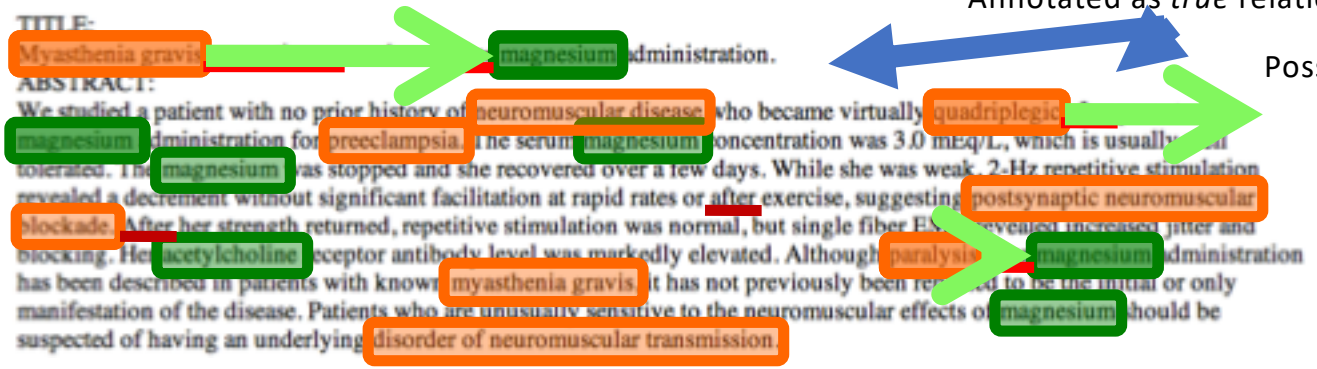
Example: Chemical-Disease Relation Extraction from Text



- 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

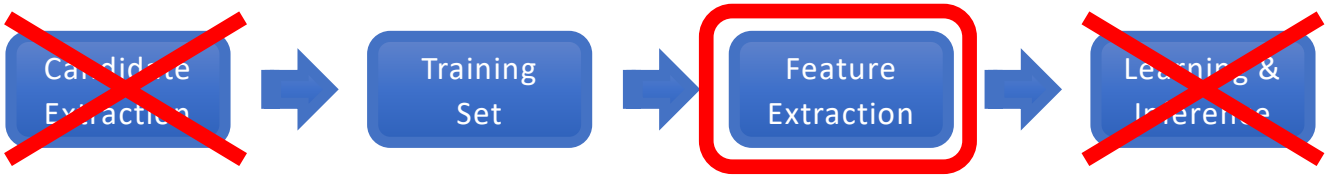
Relation Extraction with Machine Learning



Example binary features:

- `PHRASE_BTWN["presenting as"]`
- `WORD_BTWN["after"]`

TODAY:

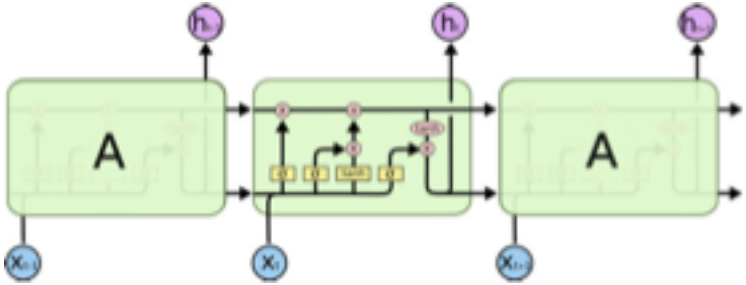


Used to be: *Feature engineering* is the bottleneck

Rise of Deep Learning

3. The BiLSTM Hegemony

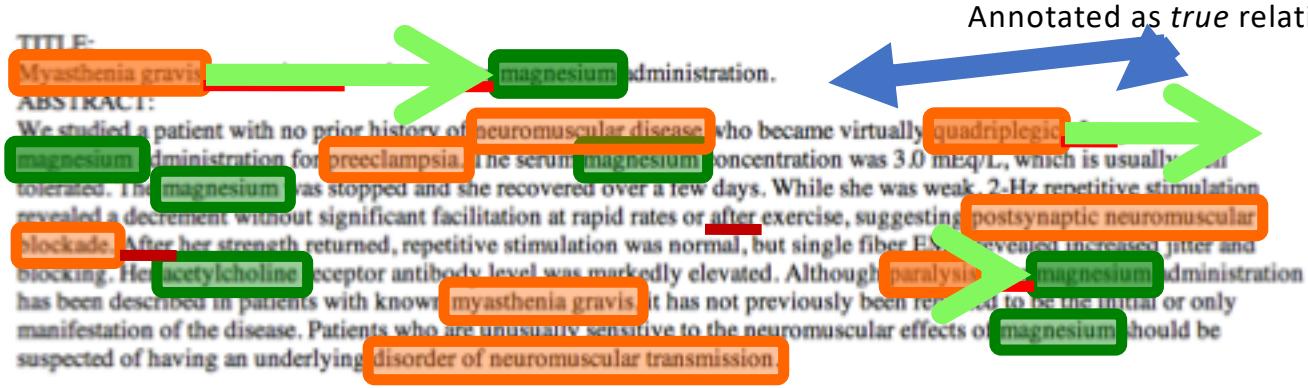
**To a first approximation,
the de facto consensus in NLP in 2017 is
that no matter what the task,
you throw a BiLSTM at it, with
attention if you need information flow**



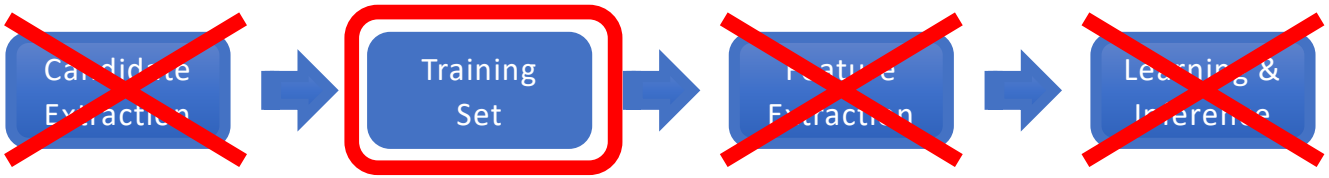
Chris Manning

Feature engineering is dying! (sort of)

Relation Extraction with Machine Learning



For a basic real-world use case:



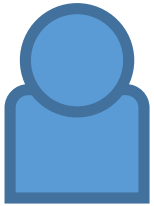
…If we have ***massive*** training sets.

KEY IDEA: Noise-aware learning

By modeling noise in training set creation **process**,

we can use *many*, **low-quality** sources to train **high-quality** models.

Programming in Snorkel

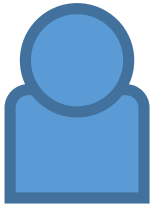


- The user
  - ***Loads in*** unlabeled data
  - ***Writes*** labeling functions (LFs)
  - ***Chooses*** a discriminative model, e.g., LSTMs

- Snorkel
  - ***Creates*** a noisy training set- *by applying the LFs to the data*
  - ***Learns*** a model of this noise- *i.e. learns the LFs' accuracies*
  - ***Trains*** a *noise-aware* discriminative model

Importantly, **no** hand-labeled training sets.

Programming in Snorkel



- The user
  - ***Loads in*** unlabeled data
  - ***Writes*** labeling functions (LFs)
  - ***Chooses*** a discriminative model, e.g., LSTMs

Main user input!

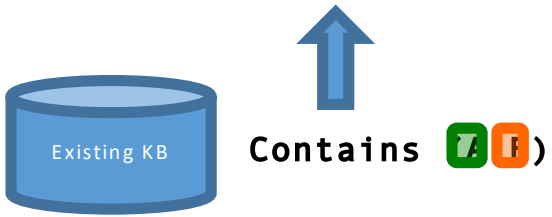
- Snorkel
  - ***Creates*** a noisy training set- *by applying the LFs to the data*
  - ***Learns*** a model of this noise- *i.e. learns the LFs' accuracies*
  - ***Trains*** a *noise-aware* discriminative model

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…”



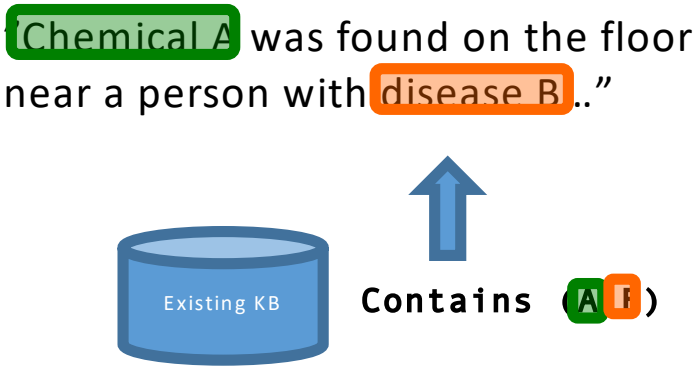
Label = TRUE

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
```



Label = TRUE

...can be false!

We learn the accuracy of the rule without training data but ***assuming there are other rules*** *(more soon)*

A Unifying Method for Weak Supervision

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

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

Need: A formal way of capturing & combining *many* sources of weak supervision. ***Our choice: generative graphical models.***

Labeling Function Dependencies

Users can also define *dependencies* between the labeling functions

```
def lf1(x):
    m = re.search(r'.*cause.*', x.between)
    return 1 if m else 0
```

```
def lf2(x):
    m = re.search(r'.*not cause.*', x.between)
    return 1 if m else 0
```

Learn *dependencies* [Bach et al. ICML 17] and *features* under which labeling functions quality varies [Varma et al. 17]—all *without* labels.

Snorkel Tidbits: Software 2.0?

- **One type of input.** *Everything is supervision.*
  - Compared to feature engineering:
    - Lighter weight, instant feedback.
    - Leverage Deep Learning models

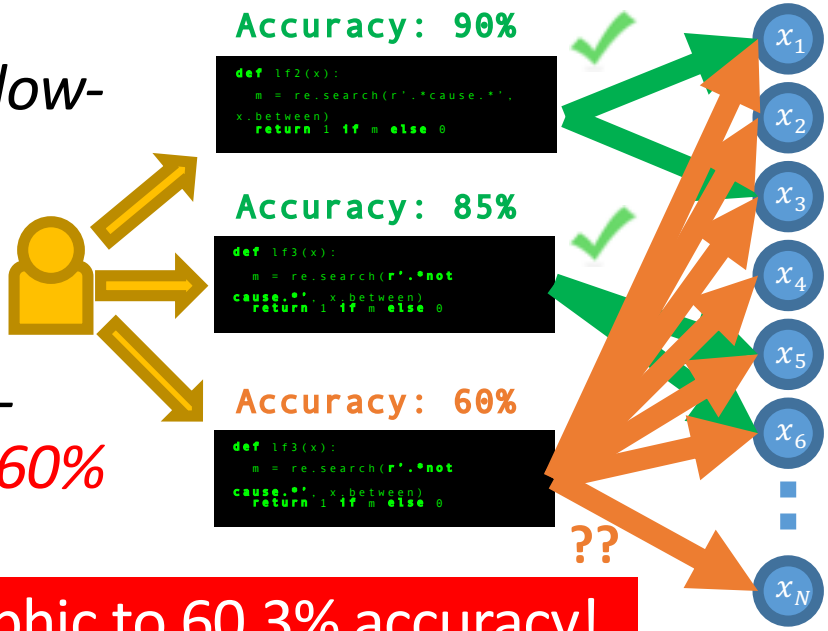
- **Productivity** Ease of high-precision rules with improved recall of modern methods.

- **Supervision as code.** Modular, debugging, etc.
  - Label reuse for new tasks
  - More unlabeled data leads to higher quality

A new challenge: *Dealing with training data of uneven quality.*

User Frustration Debugging Distant Supervision

- Wrote several *high-precision, low-coverage* distant supervision rules… *10k points @ 90%*
- … and one *low-precision, high-coverage* rule … *1M points @ 60%*

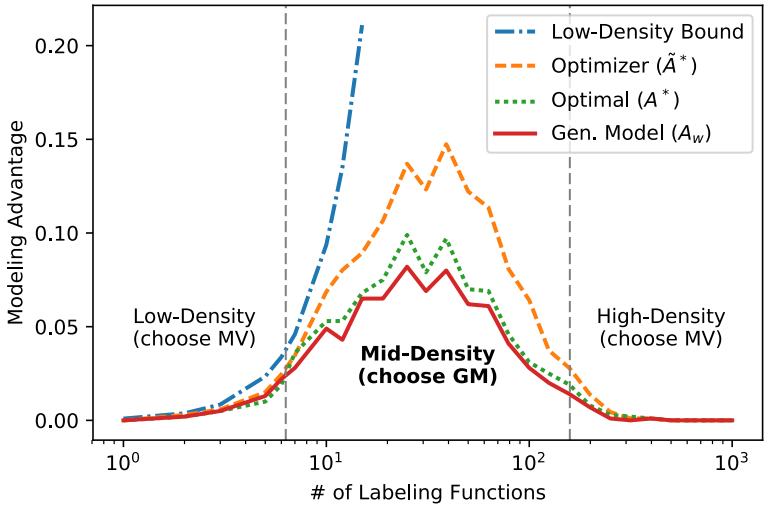


Without modeling lineage, isomorphic to 60.3% accuracy!

A very frustrating experience for users

Rough Intuition: When does modeling the noise help?

Density (labeling functions to points)



When one has moderate **density of labels** functions of **uneven** quality modeling quality improves—never hurts.

How to handle such a diversity of weak supervision sources?

Mechanisms later…

Current Users

Snorkel
Mobilize
Workshop
[7/19-20]

Snorkel is a framework for weak supervision

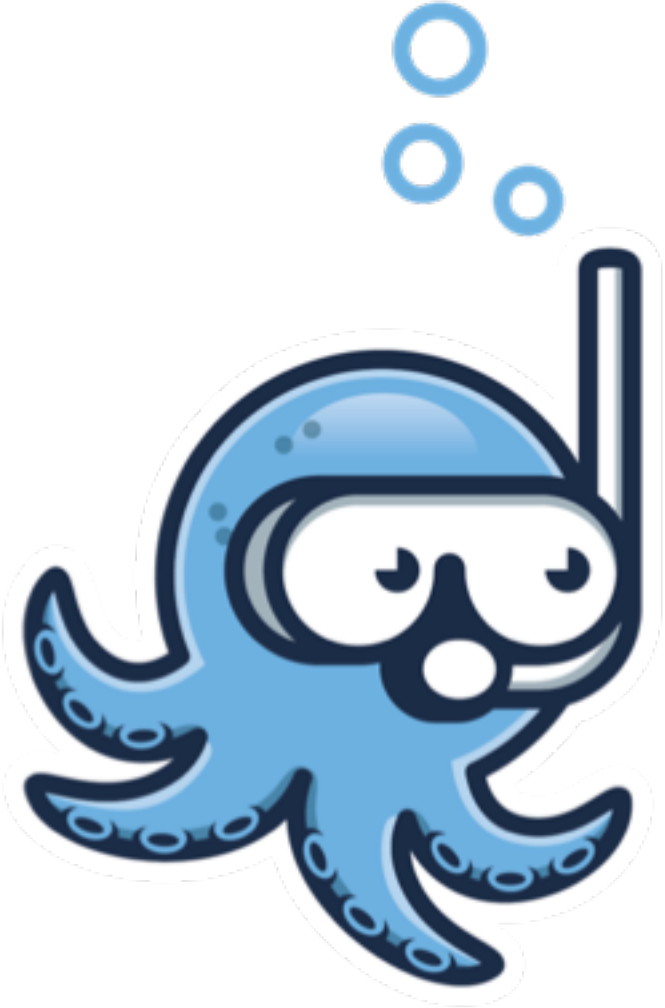
- Subsumes a wide range of weak supervision types and modeling techniques, e.g.:
  - Crowdsourcing [Dawid and Skene 1979; Karger, et. al. 2011; etc.]
  - Distant supervision [Mintz et al. 2009, etc.]
  - Ensembles of weak classifiers
  - Domain heuristics

- And related to many other types of weak supervision:
  - Label expectations / measurements [Druck, Settles, and McCallum 2009, Liang et. al. 2009]
  - Annotator rationales [Zaidan et. al. 2008]

Check out our full list @ hazyresearch.github.io/snorkel/blog/ws_blog_post.html – *send us your feedback!*

Related Work in Weak Supervision

- ***Distant Supervision:*** Mintz et. al. 2009, Alfonesca et. al. 2012, Takamatsu et. al. 2012, Roth & Klakow 2013, Augenstein et. al. 2015, etc.
- ***Crowdsourcing:*** Dawid & Skene 1979, Karger et. al. 2011, Dalvi et. al. 2013, Ruvolo et. al. 2013, Zhang et. al. 2014, Berend & Kontorovich 2014, etc.
- ***Co-Training:*** Blum & Mitchell 1998
- ***Noisy Learning:*** Bootkrajang et. al. 2012, Mnih & Hinton 2012, Xiao et. al. 2015, etc.
- ***Indirect Supervision:*** Clarke et. al. 2010, Guu et. Al. et. al. 2017, etc.
- ***Feature and Class-distribution Supervision:*** Zaidan & Eisner 2008, Druck et. al. 2009, Liang et. al. 2009, Mann & McCallum 2010, etc.
- ***Boosting & Ensembling:*** Schapire & Freund, Platanios et. al. 2016, etc.
- ***Constraint-Based Supervision:*** Bilenko et. al. 2004, Koestinger et. al. 2012, Stewart & Ermon 2017, etc.
- **Propensity SVMs:** Joachims 17



Code, Papers, Blogs, tutorials…
Feedback welcome!

Snorkel.Stanford.Edu

(More) Structured Data Applications and Less Structured Applications

HoloClean: Weakly-supervised Data Cleaning

Goal: Detect and repair errors in structured data

Diverse errors:

(i) Typos and formatting

(ii) Conflicting values

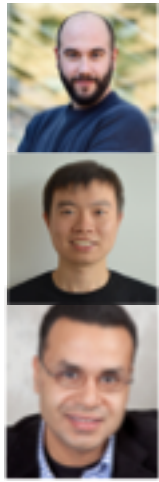
(iii) Outlier values

	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



Users provide *high-level qualitative constraints* and external data. **No other supervision required**!

HoloClean has ~ 90% precision & ~ 76% recall on real data sets—2x higher F1 score than SotA

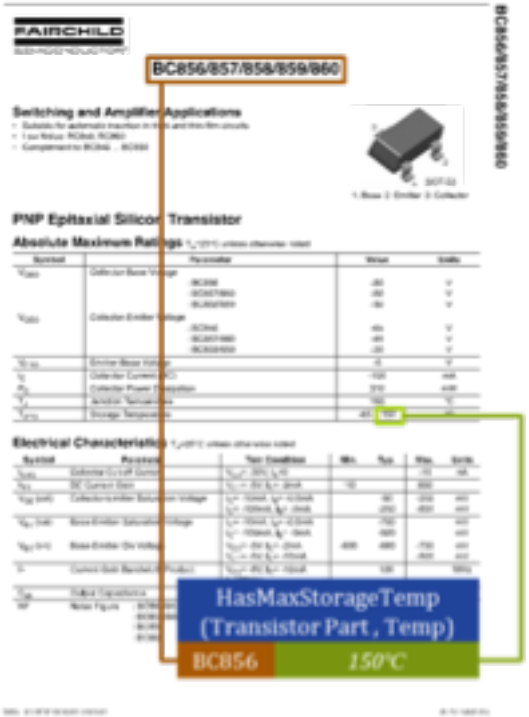
Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas C.Ré **VLDB17**

HoloClean

HoloClean released!
http://holoclean.io/

Fonduer: Handling Richly-Formatted Data

SIGMOD 2018



Challenges:

(1) Document-level relations

(2) Multimodal information

(3) Data variety

Doc. level Candidates		Multimodal Supervision: Horizontal Align with "℃"	Multimodal Supervision: Row Ngrams Gontain 'Junction'	Multimodal Supervision: Temp Value in Table
BC856	*160*	✓	✗	✓
BC856	*-65*	✓	Ø	✗
BC856	*150*	✓	Ø	✓

Data programming with labeling functions written over richly formatted data in unified data model

	Prec.	Rec.
MEMEX	87%	89%
IoT	73%	81%
GWAS	89%	81%
Paleo	72%	38%

DARPA MEMEX

SING

Macrostrat Lab

Stanford MEDICINE

Alibaba Group

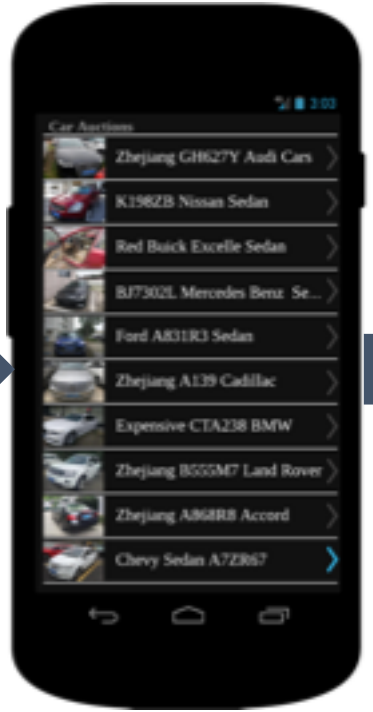
Input: User-customized HTML auction pages → **Output:** Structured knowledge base

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

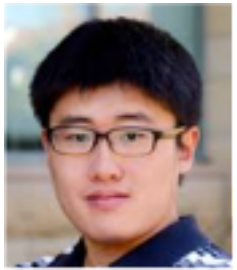
Fonduer

Alibaba.com

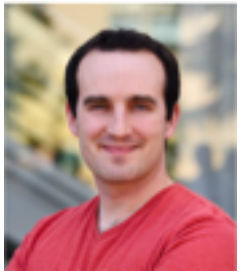
Improve auction searching quality and UX



Fonduer: Real work



Sen Wu



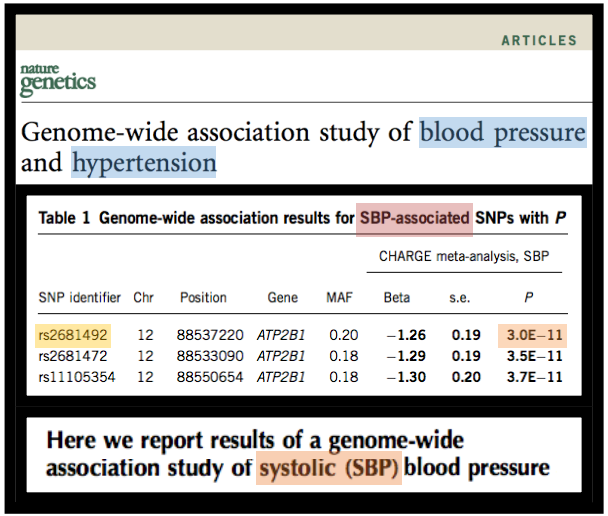
Braden Hancock

Ines Chami

Theo Retkatsinas

A Machine Compiled Database of Genome Wide Association Studies

Biomedical Publication



ARTICLES

nature genetics

Genome-wide association study of blood pressure and hypertension

Table 1 Genome-wide association results for SBP-associated SNPs with *P*

					CHARGE meta-analysis, SBP		
SNP identifier	Chr	Position	Gene	MAF	Beta	s.e.	*P*
rs2681492	12	88537220	*ATP2B1*	0.20	−1.26	0.19	3.0E−11
rs2681472	12	88533090	*ATP2B1*	0.18	−1.29	0.19	3.5E−11
rs11105354	12	88550654	*ATP2B1*	0.18	−1.30	0.20	3.7E−11

Here we report results of a genome-wide association study of systolic (SBP) blood pressure

Machine Reading

Structured Database

Variant	rs2681492
Simple phenotype	Hypertension | Blood pressure
Detailed phenotype	Systolic
p-value	3.0e-11
Source	PMID: 19430479, Tbl. 1

Database	Statistics over open-access papers	
	Associations	Unique Associations
GWAS Catalog	8,384	2,026
GWAS Central	5,914	**364**
GwasKB (ours)	6,231	**2,777**

Existing databases are incomplete

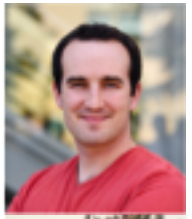
GwasKB finds 2,700 new associations

Volodymyr Kuleshov

BabbleLabble: Using Natural Language as Weak Supervision

ACL 2018

NIPS DEMO 2017



The hazards of optic nerve toxicity due to ethambutol are known.

Does this chemical cause this disease?

Yes, because the words "due to" occur between the chemical and the disease.

```
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
```

Semantic parser generates noisy LFs

Ex: In collaboration with a neuroscience startup, 30 NL explanations = same quality as ***600*** hand-labeled examples!

Vision Applications: Paroma Varma

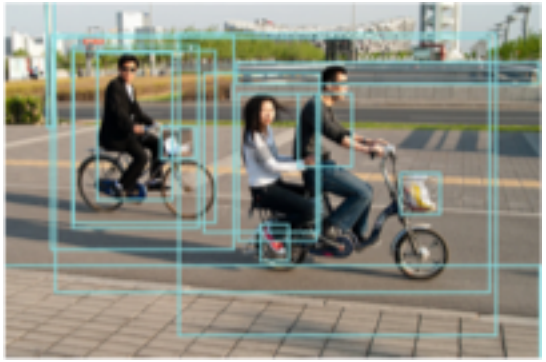
How do we write intuitive heuristic functions for images?

Problem: Need external source of information or some layer of abstraction-—Object detectors started working!

Coral: Weak Supervision Defined over Domain-Specific Primitives (DSPs)

NIPS 2017

NIPS ML4H 2017



```
return (person.y > bike.y)
```

```
return (roi.area > 1000)
```

Application	Model	Improvement Over			
		MV	Indep	Learn Dep	FS
Visual Genome	GoogLeNet	7.49*	2.90*	2.90*	-0.74*
ActivityNet	VGGNet+LR	6.23*	3.81*	3.81*	-1.87*
Bone Tumor	LR	5.17	3.57	3.06	3.07
Mammogram	GoogLeNet	4.62	1.11	0	-0.64

Approaches quality of hand-labeled training data with tens of LFs (exceeds when unlabeled data is plentiful!)

Technical Challenge: Using DSPs introduces statistical correlations…

Use static analysis to define possible correlations—open black box!

(Follow on: Synthesize the models…)

A Preview of Results...

Model	Accuracy
Fully Supervised	67.82%
Majority Vote across Heuristics	65.72%
Learn Accuracies for Heuristics	67.32%
Learn Dependencies among Heuristics	67.83%

~1000 X-rays hand-labeled by radiologists

~2000 X-rays noisily labeled by heuristics

model by analyzing the code users write!

When is Automation Helpful? Computer Vision

Is there a store front in this image?

Is there a person riding a bike in this image?

Is there a bus in this image?

Need training labels over same dataset for many tasks. Doubled size of Visual Genome with weak supervision

Ranjay Krishna

Fei-Fei Li

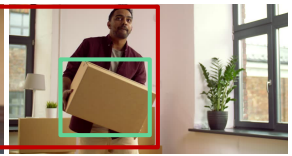
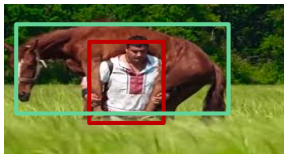
With limited labeled data, when do we use weak supervision or transfer learning?

Categorical complexity for ride:



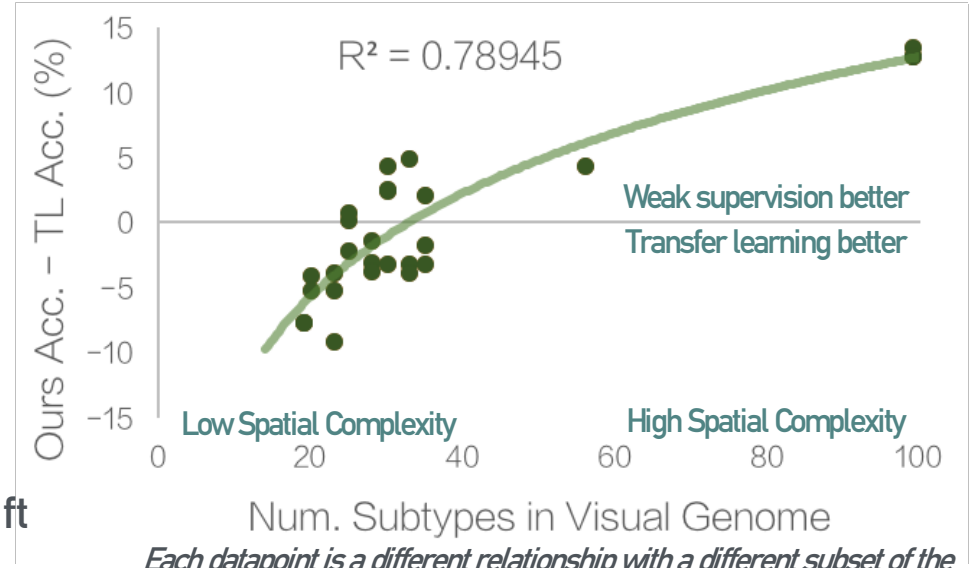

Spatial complexity for carry:




We define a complexity measure of a relationship based on spatial variance between subjects and objects. *"Unseen species estimator"*

Can spatial complexity explain when weak supervision does better than transfer learning?

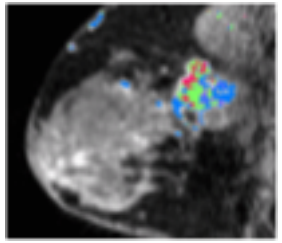


Each datapoint is a different relationship with a different subset of the examples with ground truth labels

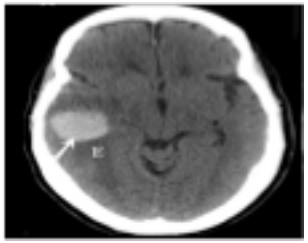
Measure spatial complexity by mean-shift clustering on primitives + inspection.

High spatial complexity correlates with gap.

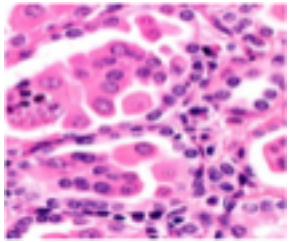
X-Modal: Expansion in Progress…



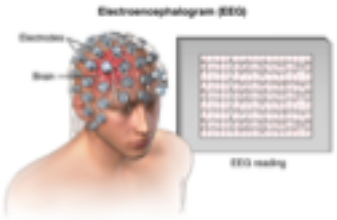
Detecting Breast Cancer on MRI



Identifying Hemorrhage on Head CT



Staging Pathology Exams



EEG (text -> time series)

EEG (RNA -> DNA)

Weak supervision isn’t just noisy labels…

Ginger Smith (CMU)

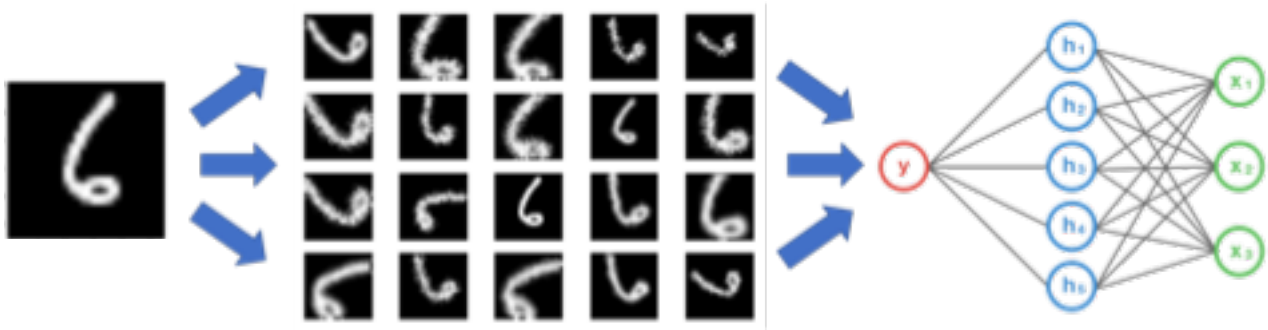
Tri Dao

Alex Ratner

Henry Ehrenberg

Data augmentation is a form of supervision

goal: improve generalization by incorporating known invariances

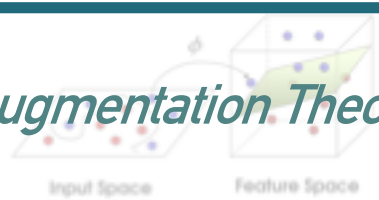


challenges: automating **the process and understanding the** *theoretical implications*

TANDA: Learning to Compose Transformations

[NIPS '17]

Augmentation Theory



[ICML Workshop '18]

Learning to Compose Domain-Specific Transformations for Data Augmentation

Alex Ratner

Henry Ehrenberg

Blog Post: snorkel.stanford.edu/blog/tanda.html

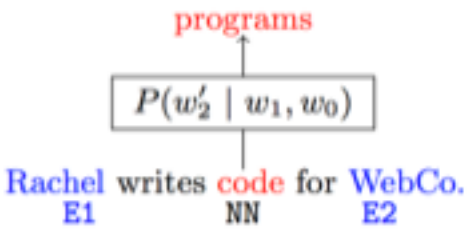
Code: github.com/HazyResearch/tanda

Data augmentation by specifying invariances

Images

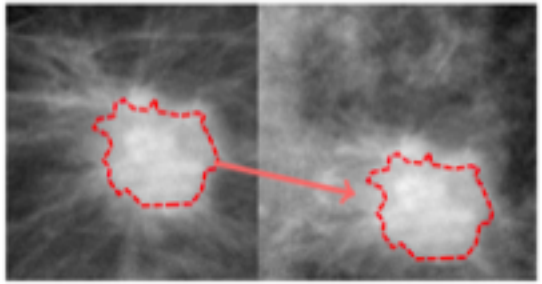
- Rotations
- Scaling / Zooms
- Brightness
- Color Shifts
- Etc…

Text



- Synonymy
- Positional Swaps
- Etc…

Medical



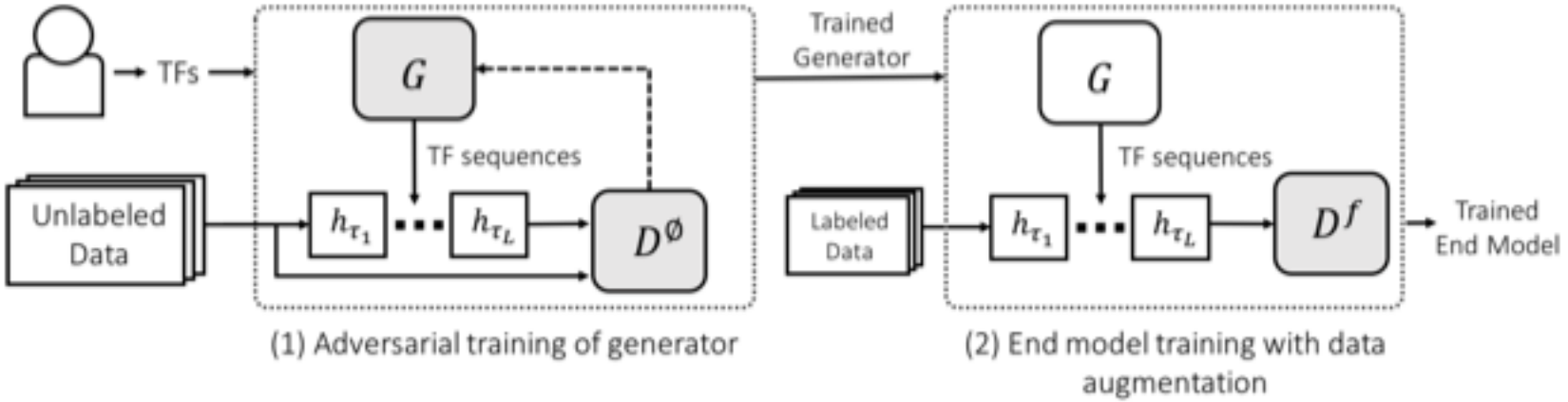
Domain-specific transformations.

Ex:

1. *Segment tumor mass*
2. *Move*
3. *Resample background tissue*
4. *Blend*

How do we choose which to apply? In what order?

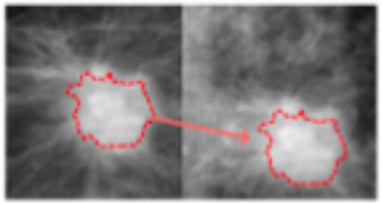
Learning to Compose Domain-Specific Transformations for Data Augmentation



Given a set of transformation primitives, we **learn sequences of transformations** using adversarial techniques

Using Complex, Domain Expert-Provided Transformations

Task	Model	None	Basic	Heuristic (Rand.)	Our Method	Difference
DDSM	9-Layer CNN	57.6	58.8	59.3	61.0	1.7
DDSM + DS				**53.7**	**62.7**	**9.0**



Learns to avoid “catastrophic” combinations

Modeling Transformation Sequences Improves Performance

Task	Model	Subsample	None	Basic	Heuristic (Rand.)	Our Method	**Improvement**
MNIST	9-Layer All-Conv CNN	10%	97.3	98.7	99.0	99.2	**0.2**
CIFAR-10	56-Layer ResNet	10%	66.0	73.1	77.5	81.5	**4.0**
		100%	87.8	91.9	92.3	94.0	**1.7**
ACE (F1 Score)	Bi-LSTM	100%	62.7	59.9	62.8	64.2	**1.4**

And, we can use arbitrary end models- simple ones above more possible!

Snorkel.Stanford.edu

Conclusion For Lecture 1

Programming is changing

Can make AI and ML techniques **radically easier**, our focus on modeling supervision process

Feel free to spit 🔥 🔥 🔥 -- we'd love your feedback and pointers to your work

snorkel

Theory

Goals in this section

1. Dive more deeply into generative model theory, so you can understand its limitations and it's intrinsically interesting.

2. Give you a taste of the problems in the machine learning side that come up.

Data Programming Pipeline in Snorkel

Input: Labeling Functions, *Unlabeled data*

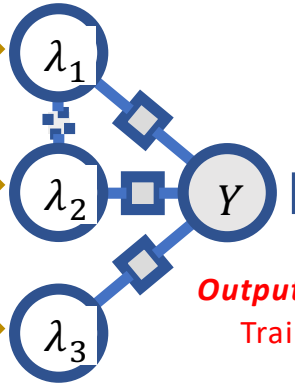
DOMAIN EXPERT

```
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
```

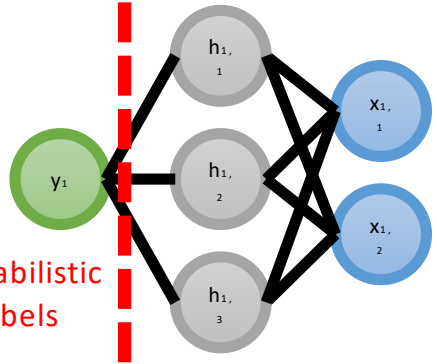
```
def lf3(x):
    m = re.search(r'.*not
cause.*', x.between)
    return 1 if m else 0
```

Generative Model

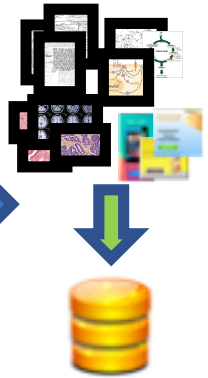


Output: Probabilistic Training Labels

Noise-Aware Discriminative Model



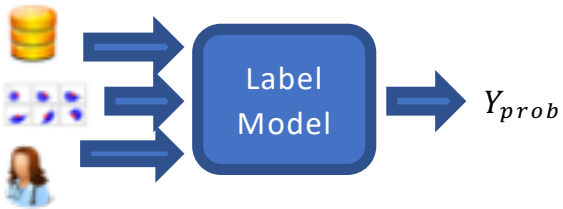
Ex. Application: Knowledge Base Creation (KBC)



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

Technical Challenge: Modeling Weak Supervision

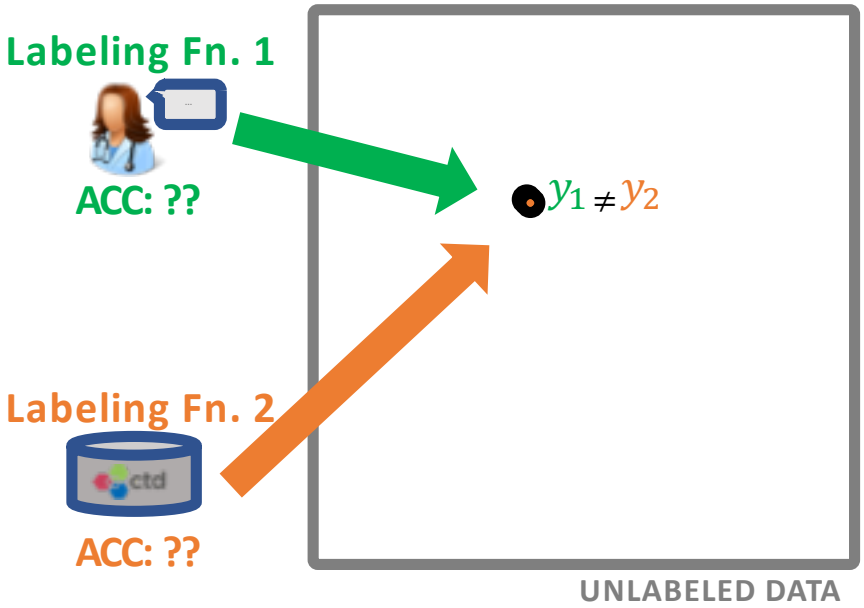
- Input: **noisy, overlapping and conflicting** labels from LFs of ***unknown* accuracies**
- Goal: learn the accuracies of the LFs, ***without labeled data***, to reweight and combine their labels
- A classic technical challenge, but with several fundamental twists…
  - Small # of LFs, each with large coverage
  - Correlated LFs
  - *Multi-task LFs*



Let's look at why this is critical…

Challenges of Weak Supervision

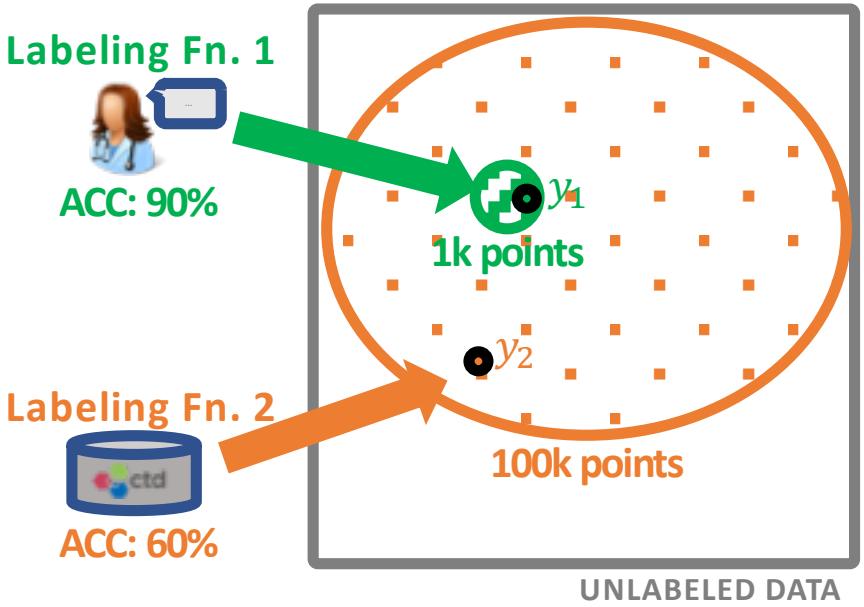
- Problem 1: How do we resolve conflicts between weak label sources?
  - How can we estimate their accuracies without ground truth?
- This is a real development burden that our users faced with prior "distant supervision" systems



Need to be able to estimate source accuracies

Challenges of Weak Supervision

- Problem 2: Need to communicate training point lineage to model being trained
- Ex:
  - User writes one high-accuracy, low-coverage LF…
  - …and one low-accuracy, high-coverage LF
  - *If we just naively take the union of labels, expected acc. = 60.3%!*



Need to communicate training label *lineage*

Data Programming Pipeline in Snorkel

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
```

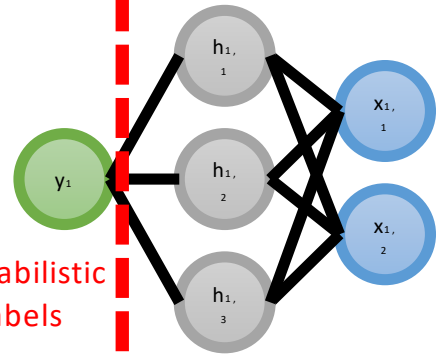
```
def lf3(x):
    m = re.search(r'.*not
cause.*', x.between)
    return 1 if m else 0
```

Label Model

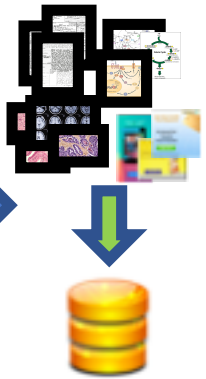
λ_1 λ_2 λ_3 Y

Output: Probabilistic Training Labels

End Model



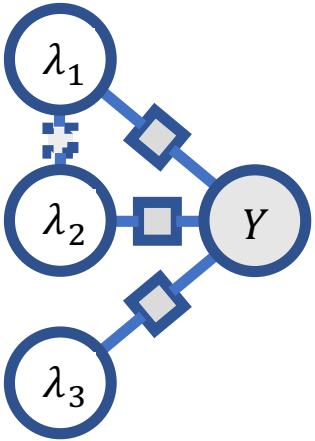
Ex. Application: Knowledge Base Creation (KBC)



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

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



Modeling LFs as Noisy Voters

- We model each labeling function as having an *accuracy* α and a *labeling propensity* β*:*

$Y \in \{-1,1\}$

$$\lambda_j(x) = \begin{cases} Y & w.p. \ \ \beta_j \alpha_j \\ -Y & w.p. \ \ \beta_j (1-\alpha_j) \\ 0 & w.p. \ \ 1-\beta_j \end{cases}$$

- And can be correlated with other LFs…

Defining basic variables of our problem

Correctness indicator variable

$$\Lambda_i = \mathbb{I}^{\pm}\{\lambda_i(x) = y\} = \lambda_i(x)y$$

Goal: Recover the accuracies

$$\alpha_i = P(\lambda_i(x) = y \,|\, y, \lambda_i(x) \neq 0)$$

Prob. i^{th} LF is correct

Given it has not abstained

$$\beta_i = P(\lambda_i(x) \neq 0)$$

Prob. i^{th} LF emits label- note this is easily estimated by empirical counts!

Let's consider $\beta_i = 1$ for exposition.

Main idea

- We can learn from the (rate) of conflicts and overlaps!
  - We can observe **agreement and disagreement.**

- **Simple experiment:**
  - Suppose that we knew labeling function 1 was exactly 0.9 accurate.
  - Labeling function 2 has an unknown accuracy, **but it** agrees 70% of examples it agrees with 1

$$\text{Then } 0.7 = \mathrm{P}(\text{Agreement}) = \ 0.9p + 0.1(1-p) = 0.8p + 0.1$$

$$\text{So } p = 0.75$$

So our idea will be to make sure they are all consistent!

Our goal: Estimate the mean of Λ_i

$$\mu_i = \mathrm{E}[\Lambda_i] = (2\alpha_i - 1)$$

Problem: We can't observe Λ_i, since it depends on the ground-truth label Y!

How do we learn the accuracies without labeled data?

Abstract: Analyzing the covariance matrix

From the definition- three terms:

$$\Sigma = \mathrm{E}[\Lambda\Lambda^T] - \underline{\mathrm{E}[\Lambda]\mathrm{E}[\Lambda]^T}$$

This is just the outer product of the vector of statistics μ

Approach: Analyzing the covariance matrix

From the definition- three terms:

$$\Sigma = \mathrm{E}[\Lambda\Lambda^T] - \mu\mu^T$$

This ends up being the matrix of pairwise overlaps between the LFs- ***which is observable!***

$$\mathrm{E}\left[\Lambda_i \Lambda_j\right] = \mathrm{E}\left[(\lambda_i(x)Y)(\lambda_j(x)Y)\right] = \mathrm{E}\left[\lambda_i(x)\lambda_j(x)\right] \coloneqq O_{i,j}$$

Approach: Analyzing the covariance matrix

We now have:

$$\Sigma = O - \mu\mu^T$$

Covariance matrix

Overlaps matrix (Observable)

Unknown means (rank-one)

The form is starting to look familiar… like a rank-one matrix approximation problem where we are fitting to the observed LF-LF agreement rates!

Warmup: Conditionally independent LFs

- To start, suppose that the Λ_i are ***independent***—i.e. the LFs make uncorrelated errors
- ➔ The covariance matrix is diagonal, meaning that for off-diagonal entries:

$$O_{i,j} = \mu_i \mu_j, \qquad \mathrm{i} \neq j$$

Symmetry: By Example

Suppose you have **three labelers** with accuracies $\alpha = (0.7, 0.8, 0.9)$

Then, $\Lambda = \alpha - (1 - \alpha)$ is the expectation of the indcators.

The observation matrix is $[(0.4, 0.6, 0.8)]$ is essentially the outer product $yy^T = V = (-y)(-y)^T$

- That is, the solution is only defined **up to sign**.
- In particular, $\alpha_0 = 1 - \alpha = (0.3, 0.2, 0.1)$ we have $\Lambda_0 = x - (1 - x) = -\Lambda$.

Can't tell if everyone is accurate—or everyone is inaccurate!

Assumption: labelers better than chance removes this symmetry
(weaker even if only the average labeler is… more laters)

Identifiability

So, when does the problem have a unique solution?

By taking log of squares of both sides, we see that:

$$\log((O_{i,j})^2) = \log(z_i^2) + \log(z_j^2), \mathrm{i} \neq j$$

We thus represent our problem as a system of linear equations, ***up to determining the signs of the z_i***s:

$$\boxed{M_\Omega l = q_\Omega,} \qquad l_i = \log(z_i^2), \qquad q_{(i,j)} = \log((O_{i,j})^2)$$

If M_Ω is invertible, we can uniquely recover the z_i up to sign.

When can we “uniquely” solve?

When is M_Ω invertible? We need more equations than unknowns!

- If there is one labeling function, hopeless!
- If two labeling functions, only one equation.
- If there are three labeling functions, then there are 3 equations!
  - What is M_Ω here?
  - Note that a **single** sign determines all signs! (why?)

In statistics, called **identifiability (more later).**

Solving as (masked) rank-one matrix approx.

- **Dealing with noise**. Note the μ are estimated from data, so have statistical noise.

$$\underset{\mu}{\operatorname{argmin}} \|O - \mu\mu^T\|_{F,i\neq j}$$

- Note the standard noise analysis gives us essentially optimal rates.
  - De Sa et ICML 15.
  - If you'd like more details, mail me (or Alex!)

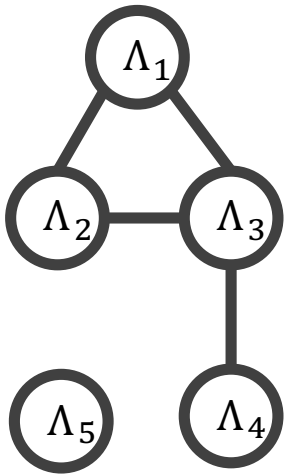
…Correlations…

Challenge

- If $\mathrm{P}(\lambda_1\lambda_2) \neq P(\lambda_1)P(\lambda_2)$. The previous section was based on this observation!

- At first glance, all hope seems lost… We need a theory to cope with this, and it is the theory of graphical models and **inverse covariance matrices.**

Correlations between LFs

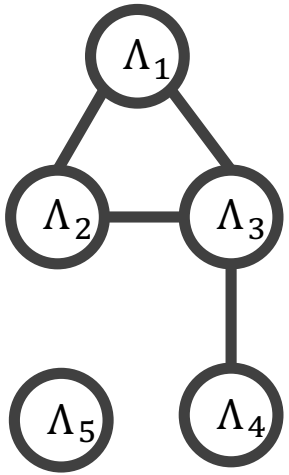


- Suppose we know the correlation **structure—but not the weights—**between the LFs
  - Can represent as a probabilistic graphical model (PGM)
  - Can handle arbitrary dependency structures!*
  - Define the edge set as Ω

- Model captured by a covariance matrix $E[(\Lambda - \mu)(\Lambda - \mu)^T]$.
  - More general correlations need tensors, but captures interesting case.

**Technical detail: For this tutorial, to simplify, we'll assume the junction tree of our dependency graph has singleton separator sets*

Handling correlations between LFs



- Suppose we know the correlation **structure—but not the weights--**between the LFs

- ***Turns out the inverse*** *covariance matrix is* ***graph structured****.* (cf. Loh & Wainwright [2013], well known for gaussians)
  - ***Fact:*** *If* λ_i , λ_j *are not connected in the graphical model, then the* $\Sigma^{-1}_{i,j} = 0$.
  - *How does this capture the indepdnent case?*

**Technical detail: For this tutorial, to simplify, we'll assume the junction tree of our dependency graph has singleton separator sets*

Using the Inverse Covariance Matrix

$$\Sigma^{-1}$$
$$= (O - \mu\mu^T)^{-1}$$
$$= O^{-1} - \frac{O^{-1}\mu\mu^T O^{-1}}{1 + \mu^T O^{-1}\mu}$$
$$= O^{-1} - zz^T$$

Sherman-Morrison formula

Define $z = \frac{O^{-1}\mu}{\sqrt{c}}$ where $c = 1 + \mu^T O^{-1}\mu$

- Now, since $\Sigma^{-1}$is graph-structured (entries not corresponding to edges in the dependency graph are zero), we have:

$$O_{i,j}^{-1} = z_i z_j, \qquad (\mathrm{i}, j) \notin \Omega$$

Once again: Can solve as (masked) rank-one matrix approx. problem

$$\underset{z}{\operatorname{argmin}} \|O^{-1} - zz^T\|_{(\mathrm{i,j})\notin\Omega}$$

- However, now it's less clear- is this problem identifiable?

Identifiability: Same trick!

- We need a compiler-like check to tell us what dependency structures lead to unique solutions… and what extra information is needed
- By taking log of squares of both sides, we see that:

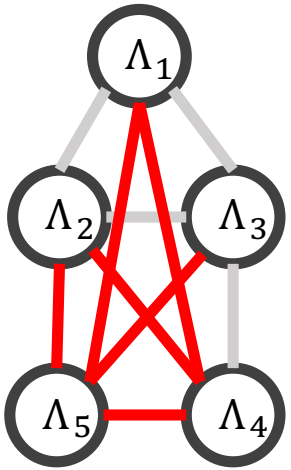
$$\log((O_{i,j}^{-1})^2) = \log(z_i^2) + \log(z_j^2), \qquad (i,j) \notin \Omega$$

- We thus represent our problem as a system of linear equations, ***up to determining the signs of the z_is***:

$$\boxed{M_\Omega l = q_\Omega,} \qquad l_i = \log(z_i^2), \qquad q_{(i,j)} = \log((O_{i,j}^{-1})^2)$$

If M_Ω is invertible, we can uniquely recover the z_i up to sign…

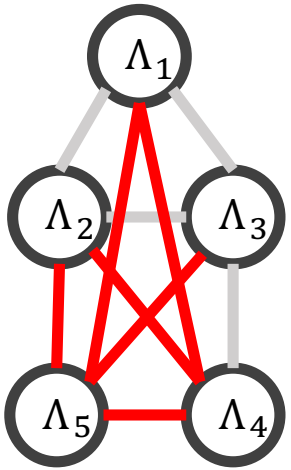
Identifiability: Determining the Signs



- For any *connected component in the inverse dependency graph* Ω_{inv}, knowing the sign of one z_i implies all the others…
- ➔We just need to know the ***sign*** of one of the z_is for each component
- If Ω_{inv} is fully-connected (see example to left), we just pick the solution where the LFs on average are non-adversarial!

If M_Ω is invertible, and we know the sign of one z_i per connected component of Ω_{inv} , then is identifiable

Said another way:



If we're willing to assume **all** labelers are better than chance, the model is (essentially) always uniquely defined.

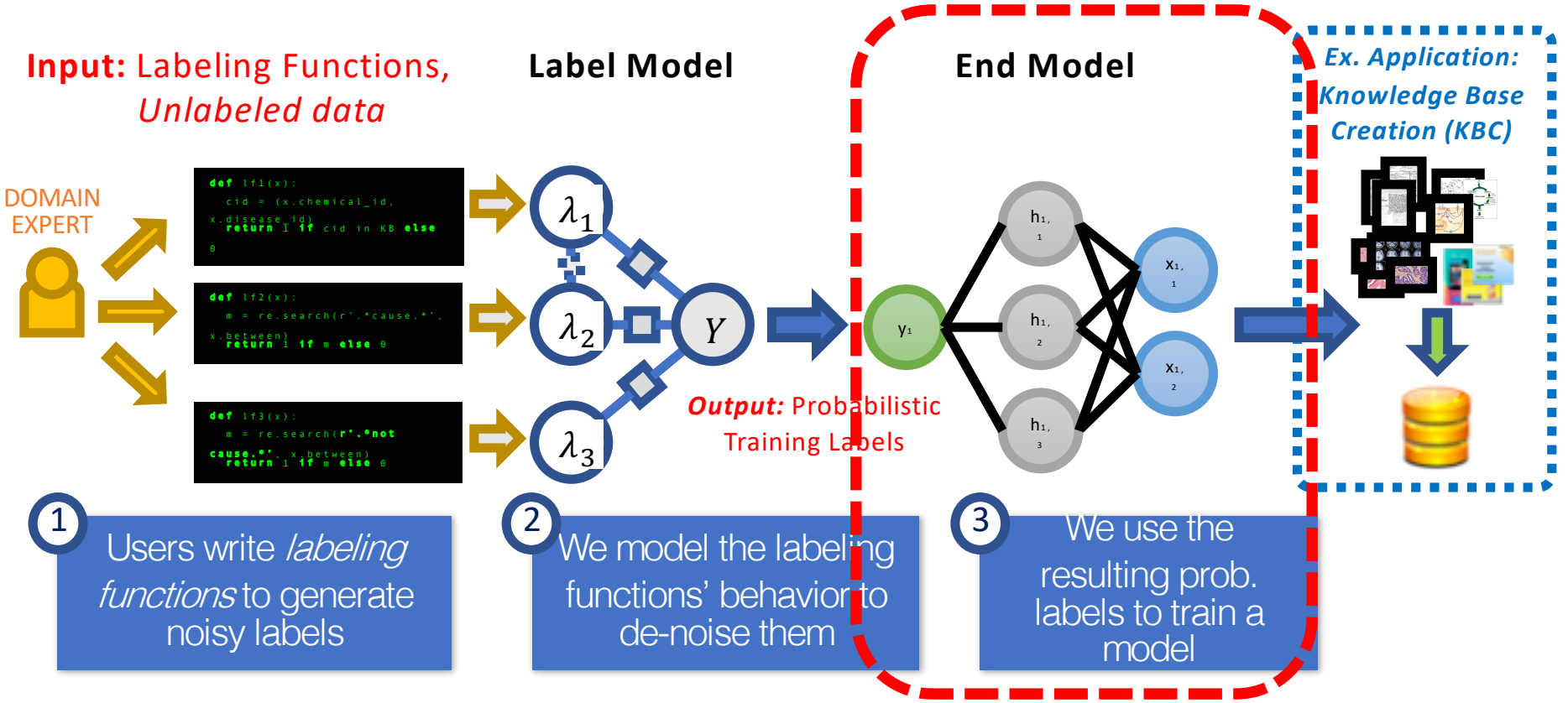
Extension: Learning the Structure of Correlations with Robust PCA

- We can also jointly estimate the LF accuracies and correlation structure → this can be phrased as instance of ***robust PCA***

$$\underbrace{O^{-1}}_{\text{Observed matrix}} = \underbrace{ZZ^T}_{\text{Low-rank matrix}} + \underbrace{\Sigma^{-1}}_{\text{Sparse matrix}}$$

We can learn the correlation structure too!

Data Programming Pipeline in Snorkel



Training a Noise-Aware Model

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

$$\hat{w} = \text{argmin}_w \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \sigma_t \, l_t(w, x^{(i)}, y_t{}^{(i)})$$

Here, we learn from the *noisy* labels:

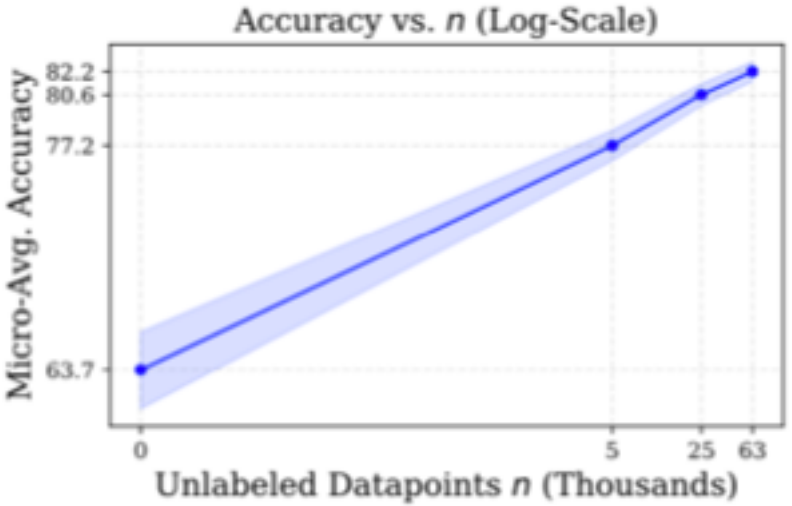
$$\hat{w} = \text{argmin}_w \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \sigma_t \, \mathbb{E}_{y \sim \boldsymbol{p_\theta}}[\boldsymbol{l_t}(w, x^{(i)}, y)]$$

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

Theoretical Result I: Scaling with Unlabeled Data

- Theory:
  - Given at least three LFs
  - And dependent on the number of LFs and the structure of their dependencies
  - ➔ end model accuracy should scale with # of *unlabeled* data points n, at same asymptotic rate as in supervised setting!

- Empirically: We indeed see this scaling with unlabeled data!

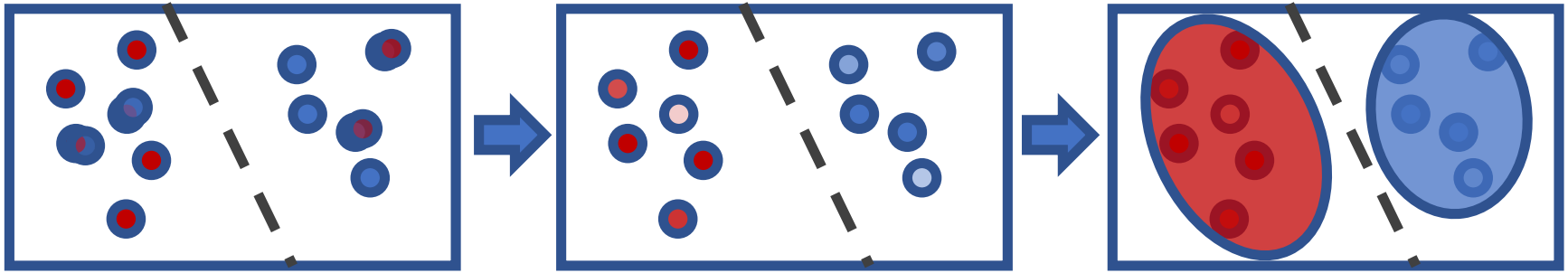


Goal: Training End Model to Generalize

Input: Labeling Functions,
Unlabeled data

Label Model

End Model



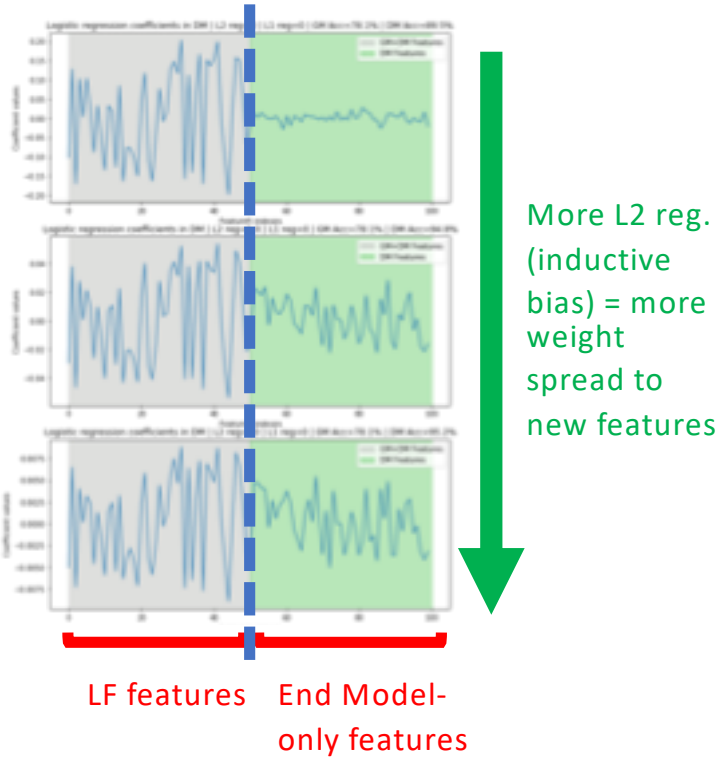
Noisy, conflicting labels

Resolve conflicts,
re-weight & combine

Generalize beyond the
labeling functions

Theoretical Results II: End model generalization

- Key question: Why should we expect the end model to generalize beyond the LFs?
- One key idea: **Inductive bias of the end model**
  - Ex: L2 reg. = bias to spread mass across many features (see right)
  - Ex: Semantic generalization via e.g. pre-trained word embeddings
  - Etc.
- Theory results here coming soon!



Summary of Theory.

We examined key theory questions: can you estimate the accuracy of labeling functions **without** labeled data?

- We examined independent and correlated cases.
- We didn't cover that you get nearly optimal noise rates!

We didn't cover proving **why** you generalize in this two-phased approach.

- Only solved in rudimentary models. Fascinating topic!

Activity Goals

- Hands-on experience with a generative and (Simple) discriminative model!

- There are also more advanced tutorials on annotation, images, etc.
  - If you finish early, try these out!