Data Creation and Training with Weak Supervision

Daeyoung Hong
Seoul National University

Introduction

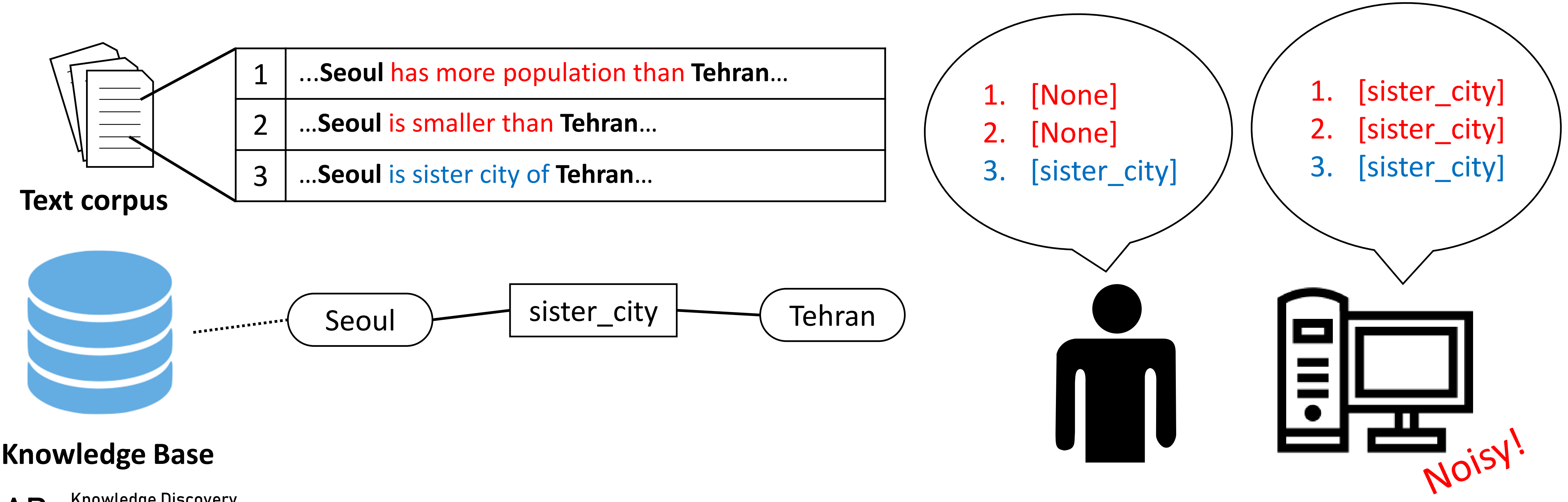
- Labeling training data is the main bottleneck in ML
  - Hiring experts to label large training data is time consuming and expensive

Introduction

- Labeling training data is the main bottleneck in ML
  - Possible solutions when we have limited resources
    - Active learning (Sequentially choose what to label among all data)
    - Transfer learning (Utilize a trained model in a domain to another domain)
    - Semi supervised learning (Utilize unlabeled data and labeled data together)
    - Weak supervision (Label unlabeled data and utilize them for training)
      - Distant supervision (machine generated labels using knowledge bases)
      - Crowdsourced labels
      - Rules and heuristics

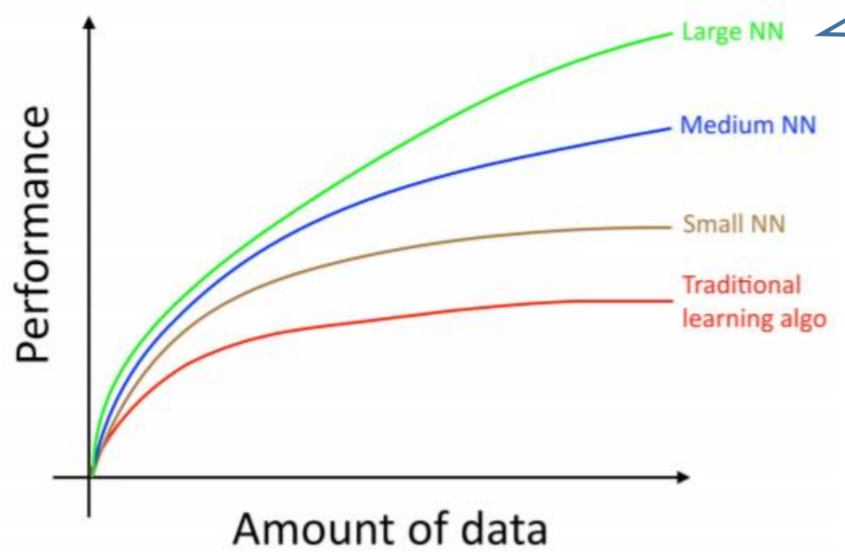
An Example of Weak Supervision

- Distant supervision quickly generates noisy labels using external data source
  - A task to find a relationship between two entities in the sentence
  - Assume that knowledge base has a relation (Seoul, *sister_city*, Tehran)
  - A distant supervision method simply annotates *sister_city* relation if there exist Seoul and Tehran entities in a sentence



Why Weak Supervision?

- A large amount of data improves the accuracy of models
- But, labeling a lot of data from human experts is expensive or practically impossible
- Weak supervision can generate a lot of labeled data quickly
- Recently, many companies and organizations have succeeded in improving model performance through weak supervision even if the generated labels are noisy



More data, better performance

[Andrew Ng. Machine Learning Yearning, Chapter 4. Scale drives machine learning progress]

Snorkel's Collaborators

- Snorkel is the most popular weak supervision project

USERS & SPONSORS

List of Papers Covered in this Seminar

- **Snorkel** [Ratner, Bach, Ehrenberg, Fries, Wu and Ré: PVLDB 2017]
  - Snorkel reduces the human effort in training data labeling by utilizing labeling functions (LFs) **without human-labeled data**
- **Snuba** [Varma and Ré: PVLDB 2018]
  - Snuba automatically generates a set of LFs by **using a small set of human-labeled data**
- **GOGGLES** [Das, Chaba, Wu, Gandhi, Chau and Chu: SIGMOD 2020]
  - GOGGLES is a domain-agnostic method for automated **image data** labeling with the affinity scores of instance pairs and **a small set of human-labeled data**
- **SPamCo** [Fan Ma, Deyu Meng, Xuanyi Dong, Yi Yang: JMLR 2020]
  - Aggregate existing models' outputs to **generate pseudo-labels** and **update models using the pseudo-labels**
- **Dual supervision framework** [Jung and Shim: COLING 2020]
  - It effectively **utilizes both weakly-supervised and human-annotated data** to train a relation extraction model

List of Papers Covered in this Seminar

- **Snorkel** [Ratner, Bach, Ehrenberg, Fries, Wu and Ré: PVLDB 2017]
  - Snorkel reduces the human effort in training data labeling by utilizing labeling functions (LFs) **without human-labeled data**
- **Snuba** [Varma and Ré: PVLDB 2018]
  - Snuba automatically generates a set of LFs by **using a small set of human-labeled data**
- **GOGGLES** [Das, Chaba, Wu, Gandhi, Chau and Chu: SIGMOD 2020]
  - GOGGLES is a domain-agnostic method for automated **image data** labeling with the affinity scores of instance pairs and **a small set of human-labeled data**
- **SPamCo** [Fan Ma, Deyu Meng, Xuanyi Dong, Yi Yang: JMLR 2020]
  - Aggregate existing models' outputs to **generate pseudo-labels** and **update models using the pseudo-labels**
- **Dual supervision framework** [Jung and Shim: COLING 2020]
  - It effectively **utilizes both weakly-supervise**
    relation extraction model

These papers propose a method to generate pseudo-labels

List of Papers Covered in this Seminar

- **Snorkel** [Ratner, Bach, Ehrenberg, Fries, Wu and Ré: PVLDB 2017]
  - Snorkel reduces the human effort in training data labeling by utilizing labeling functions (LFs) **without human-labeled data**
- **Snuba** [Varma and R
  - Snuba automatical … n-labeled data
- **GOGGLES** [Das, Cha
  - GOGGLES is a dom … ng with the affinity scores of in
- **SPamCo** [Fan Ma, Deyu Men … Dong, Yi Yang: JMLR 2020]
  - Aggregate existing models' o … ts to **generate pseudo-labels** and **update models using the pseudo-labels**

This paper proposes a method to train a model with both human-annotated and pseudo-labeled dataset

- **Dual supervision framework** [Jung and Shim: COLING 2020]
  - It effectively **utilizes both weakly-supervised and human-annotated data** to train a relation extraction model

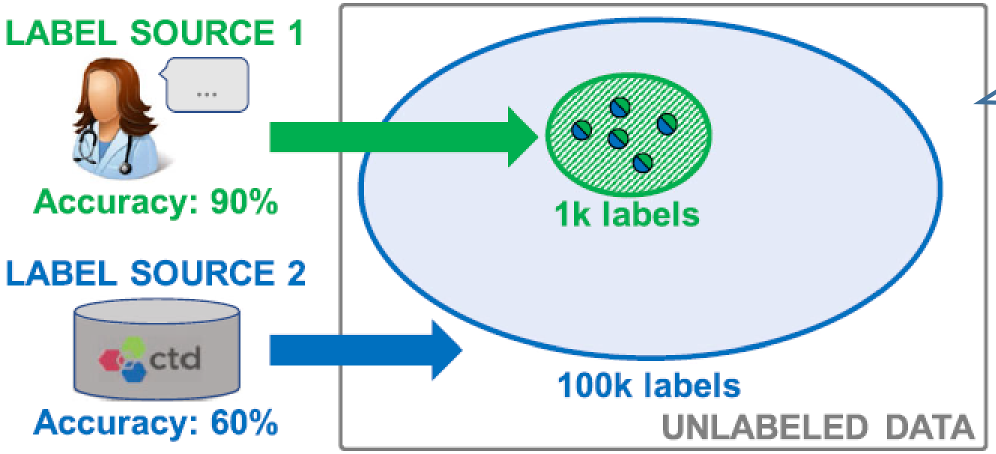
Snorkel: Rapid Training Data Creation with Weak Supervision

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries,
Sen Wu, Christopher Ré

PVLDB 2017

Various Sources of Weak Supervision

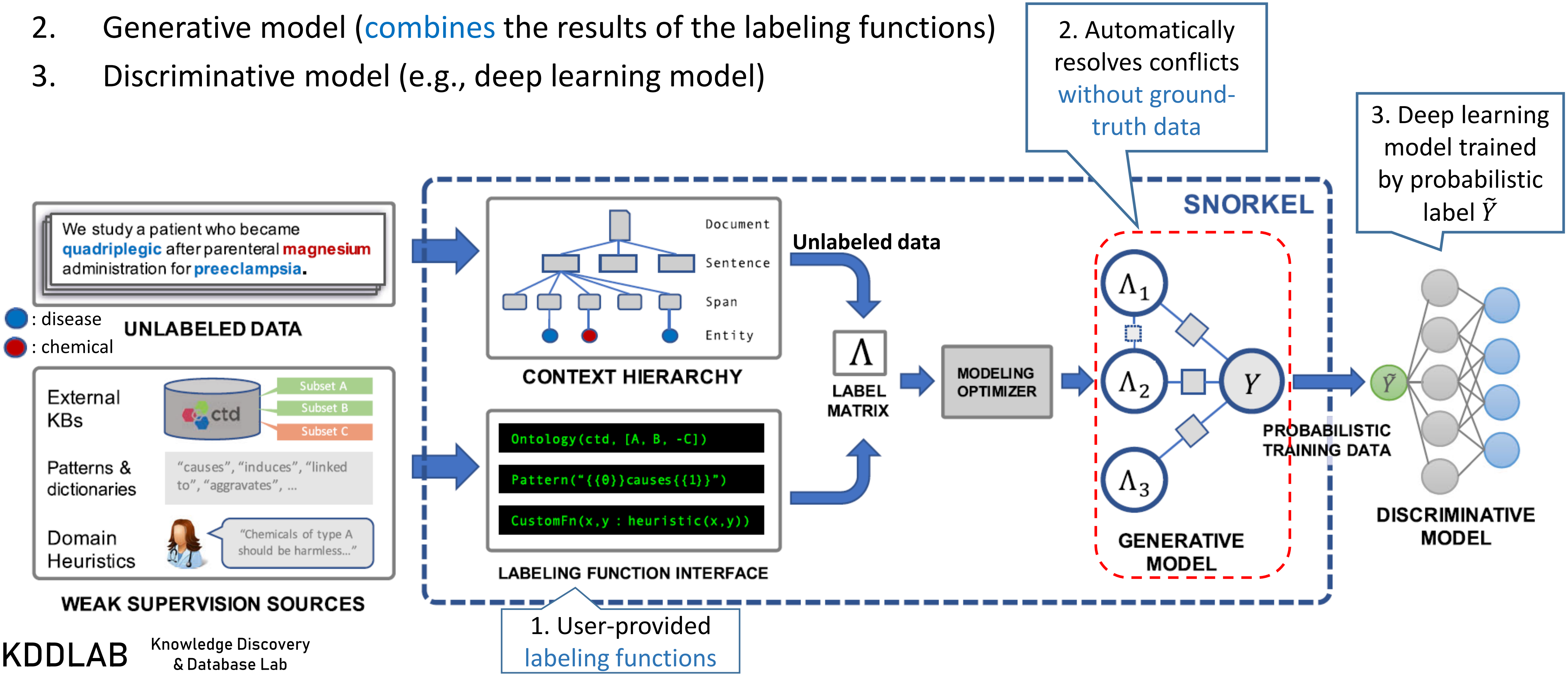
- Label source 1
  - e.g., Heuristics / pattern generated from experts
  - More accurate & low coverage
- Label source 2
  - e.g., Distant supervision
  - Less accurate & high coverage
- …



Need to resolve conflicts between label sources

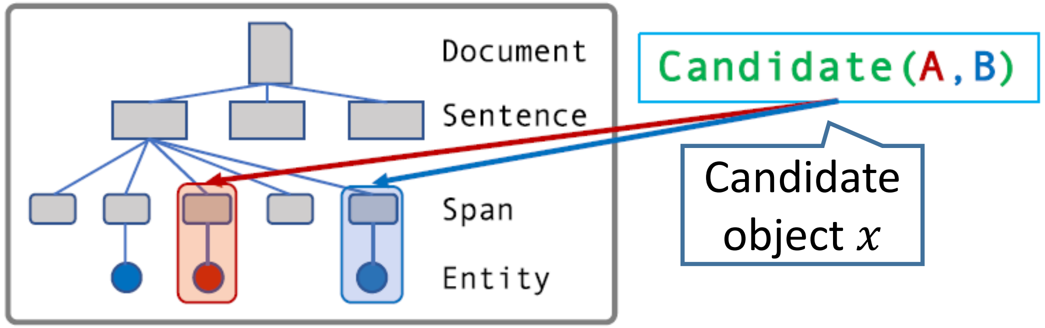
Overview of Snorkel

1. Labeling functions (rather than labeling training data)
2. Generative model (combines the results of the labeling functions)
3. Discriminative model (e.g., deep learning model)



Labeling Functions

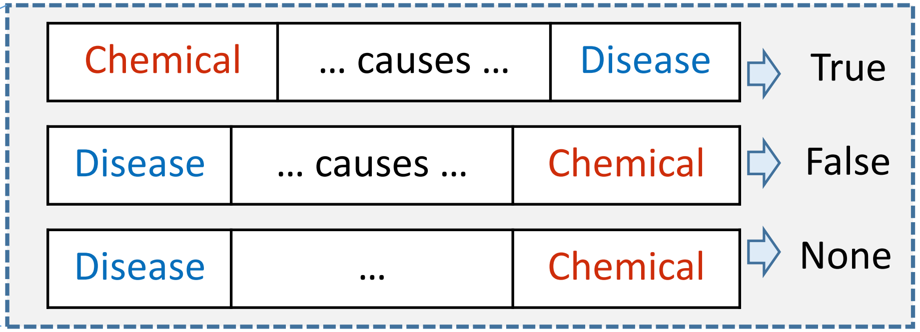
- Hand-defined labeling function
  - Arbitrary snippet of code
  - External sources (e.g., knowledge bases) can be utilized
  - input: a candidate object x with a local context
  - output: a label (True or False) or abstains (None)



Context hierarchy

```
def LF_causes(x):
  cs, ce = x.chemical.get_word_range()
  ds, de = x.disease.get_word_range()
  if ce < ds and "causes" in x.parent.words[ce+1:ds]:
      return True
  if de < cs and "causes" in x.parent.words[de+1:cs]:
      return False
  return None
```

Labeling function



e.g., If {Chemical} appears ahead of {Disease} and "causes" is between {Chemical} and {Disease}, the labeling function outputs true.

Generative Model

- **Given**
  - Label matrix Λ
    - $\Lambda_{i,j} = \lambda_j(x_i)$ (x_i: i-th unlabeled data, λ_j: j-th labeling function)
  - Generative model (GM)
    - $p_w(\Lambda, Y) = Z_w^{-1}\exp(\sum_{i=1}^{m} w^T \phi_i(\Lambda, y_i))$
      - Y: a latent variables for true labels
      - w: a vector of model parameters
      - Z_w: normalizing constant
      - ϕ_i: feature vector
        - Labeling: $\phi_{i,j}^{Lab}(\Lambda, Y) = \mathbb{I}\{\Lambda_{i.j} \neq \emptyset\}$
        - Accuracy: $\phi_{i,j}^{Acc}(\Lambda, Y) = \mathbb{I}\{\Lambda_{i,j} = y_i\}$
        - Correlation: $\phi_{i,j,k}^{Corr}(\Lambda, Y) = \mathbb{I}\{\Lambda_{i,j} = \Lambda_{i,k}\}$
- **Find**
  - Model parameters w which maximizes the marginalized likelihood
    - $\hat{w} = \mathrm{argmin}_w - \log \sum_Y p_w(\Lambda, Y)$
  - Probabilistic training labels $\tilde{Y}$
    - $\tilde{Y} = p_{\hat{w}}(Y|\Lambda)$

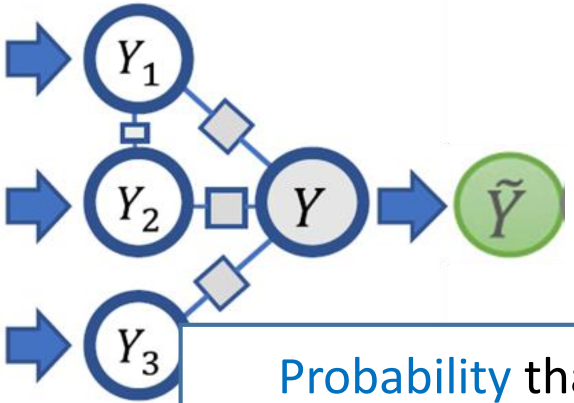
The generative model combines the results of the labeling functions

Label matrix Λ

	λ_1	λ_2	…	λ_n
x_1	1	-1		0
x_2	0	1		1
x_3	-1	0		-1
…			…	
x_m	1	0		0

1 : True
0 : Ø
-1: False

Generative model



Probabilistic labels $\tilde{Y}$

	$\tilde{Y}$
x_1	0.5
x_2	0.9
x_3	0.1
…	…
x_m	0.7

Probability that unlabeled data is true

Discriminative Model

- Instead of directly using pseudo-labels generated from the generative model, they additionally produce a discriminative model for final labeling
- The discriminative model h_θ generalizes beyond the information expressed in the labeling functions
- The model can be trained by minimizing the expected loss from the probabilistic label $\tilde{Y}$

- Noise-aware expected loss
  - $\sum_{i=1}^{m} \mathbb{E}_{y\sim\tilde{Y}}[l(h_\theta(x_i), y)]$
    - $\mathbb{E}_{y\sim\tilde{Y}}[l(h_\theta(x_i), y)] = \tilde{y}_i \cdot \boxed{l(h_\theta(x_i), 1)} + (1-\tilde{y}_i) \cdot \boxed{l(h_\theta(x_i), -1)}$
  - $\hat{\theta} = \mathrm{argmin}_\theta \sum_{i=1}^{m} \mathbb{E}_{y\sim\tilde{Y}}[l(h_\theta(x_i), y)]$

The probability of i-th unlabeled data to be **true**

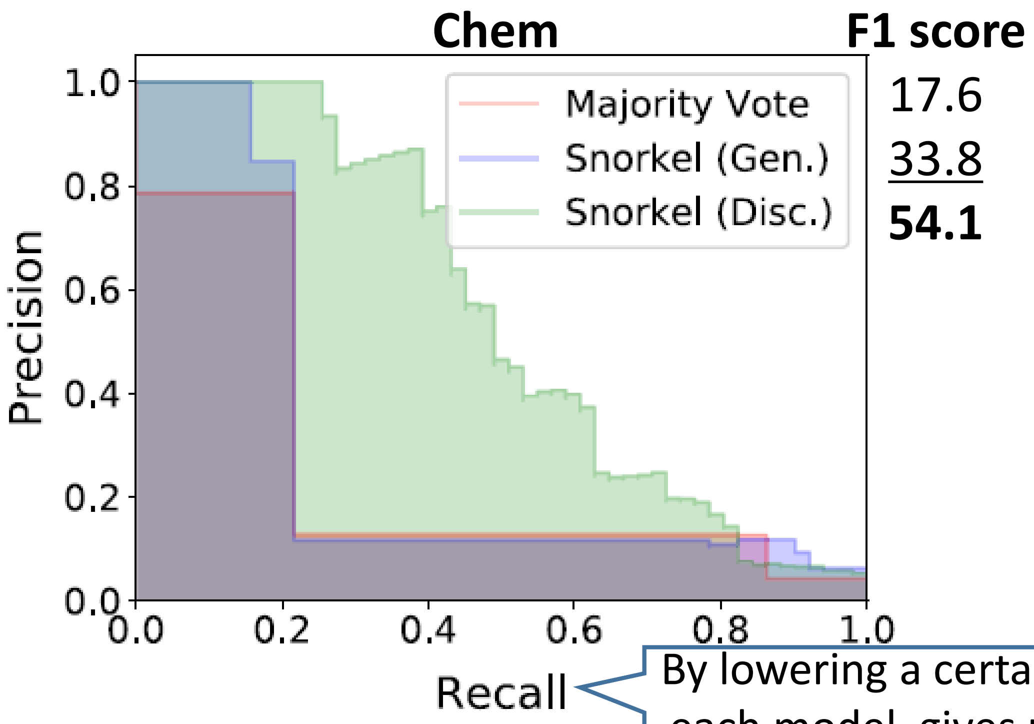
The probability of i-th unlabeled data to be **true**

The loss when the label is **true**

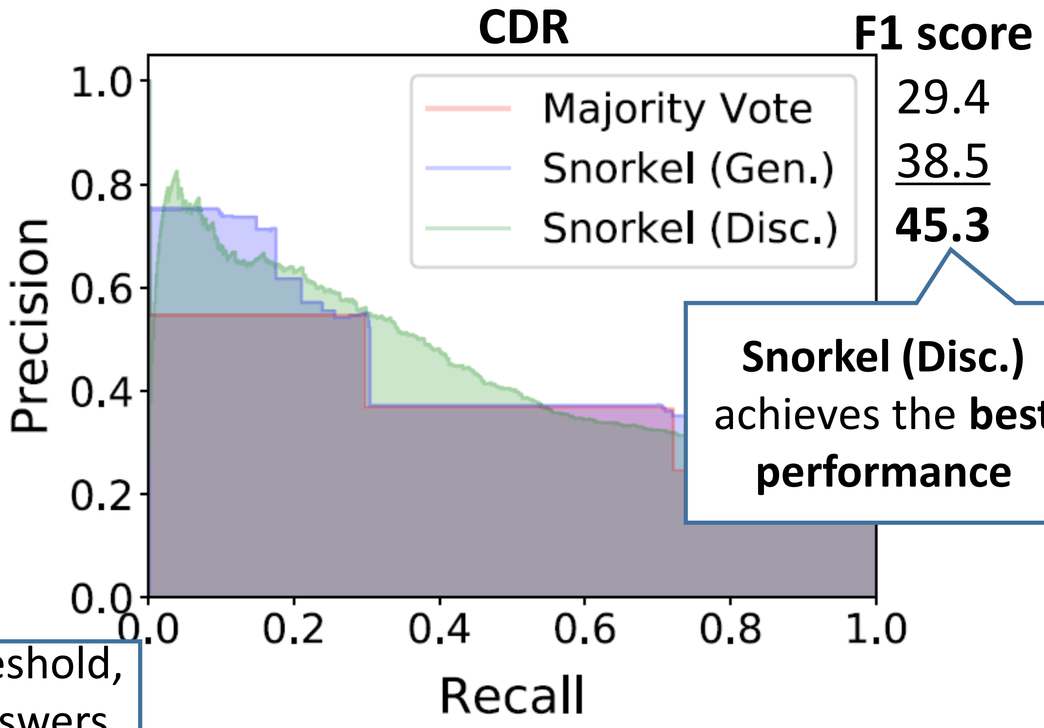
The loss when the label is **false**

Evaluation

- Models
  - **Majority Vote**: a majority vote from label functions
  - **Snorkel (Gen.)**: the generative model of Snorkel
  - **Snorkel (Disc.)**: a discriminative model trained by probabilistic labels $\tilde{Y}$ (from the generative model)
- Relation extraction tasks
  - **Chem**: extracting chemical reactions (collaborated with FDA)
  - **CDR**: finding chemical-disease relations



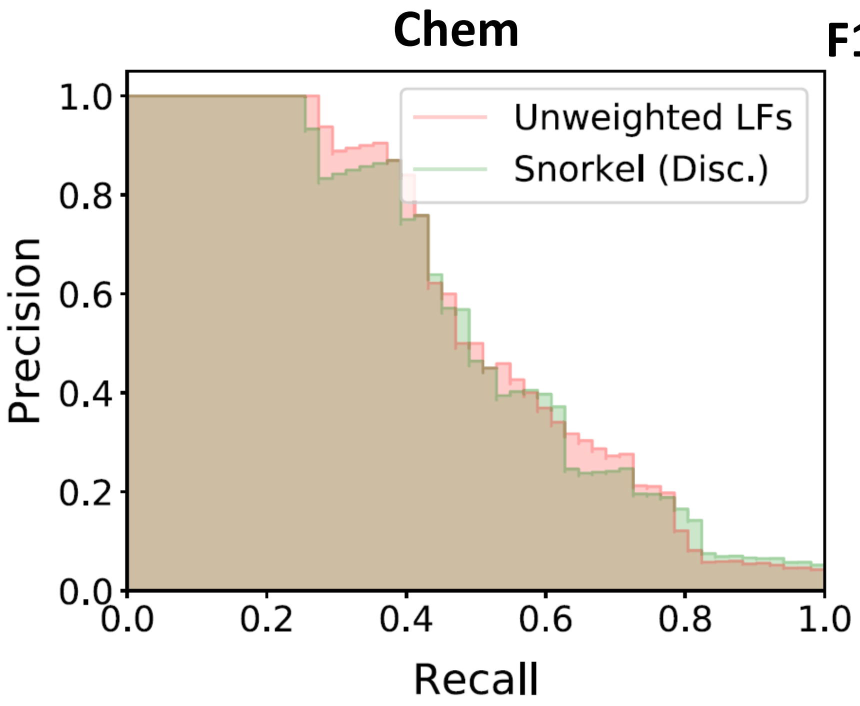
By lowering a certainty threshold, each model gives more answers but less accurate



Snorkel (Disc.) achieves the **best performance**

Effects of Generative Model

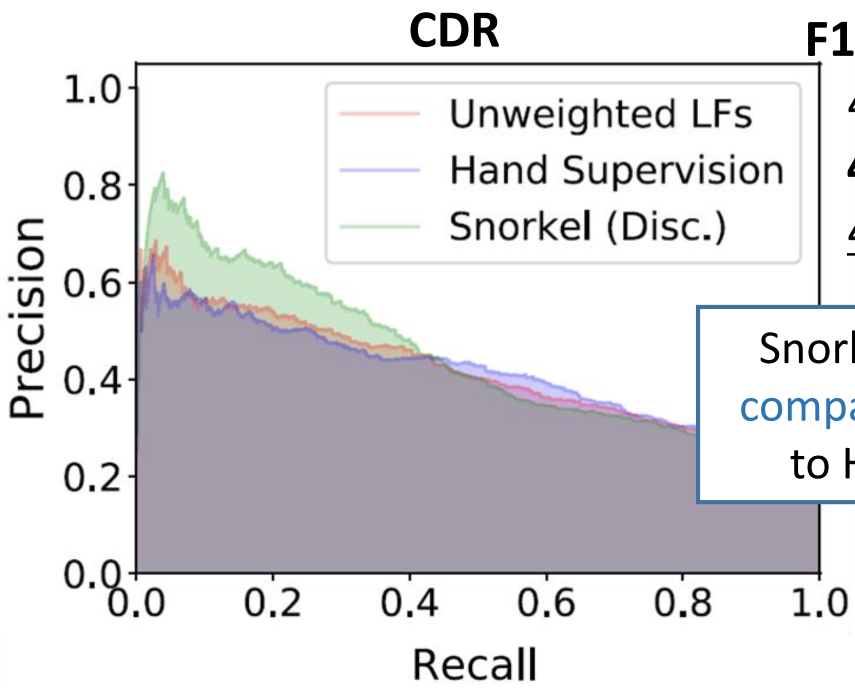
- Compared discriminative models
  - Unweighted LFs: trained by a majority vote from labeling functions
  - Hand Supervision: trained by true labels by human experts (with a lot of labor)
  - Snorkel (Disc.): trained by probabilistic labels $\tilde{Y}$ (from the generative model)



F1 score

48.6

54.1



F1 score

42.0

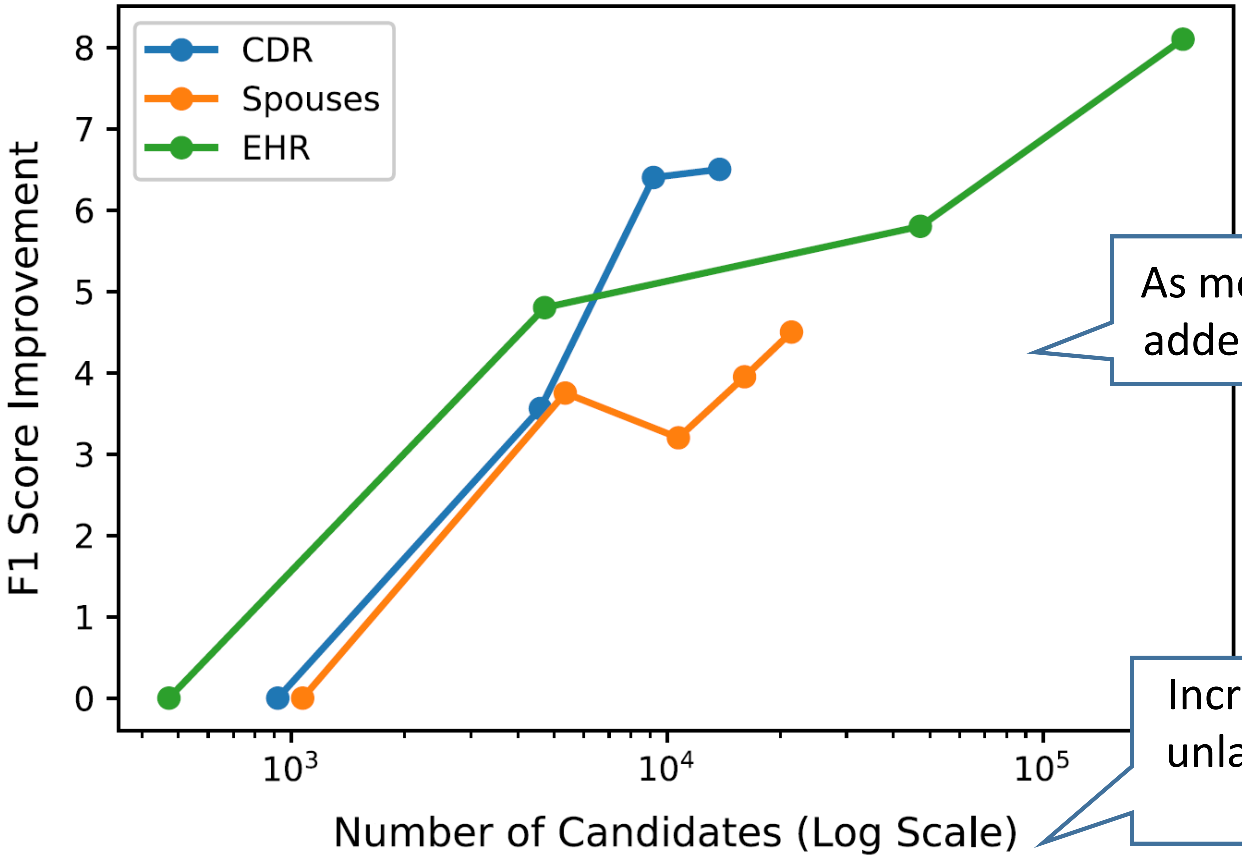
47.3

<u>45.3</u>

Snorkel (Disc.) achieves comparable performance to Hand Supervision

Additional Unlabeled Data

- Relation extraction tasks
  - **CDR**: finding chemical-disease relations
  - **Spouses**: identifying the spouse relationship between two person mentions
  - **EHR**: extracting mentions of pain levels at precise anatomical locations in electronic health records



As more unlabeled data is added, F1 score increases

Increasing the number of unlabeled data with fixed labeling functions

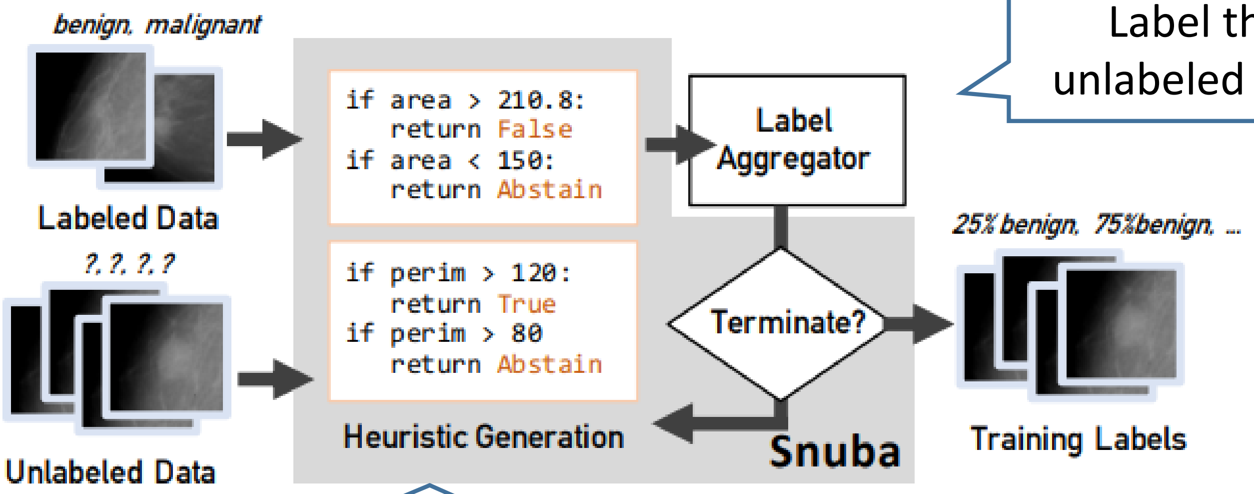
Snuba: Automating Weak Supervision to Label Training Data

Paroma Varma, Christopher Ré

PVLDB 2018

Snuba

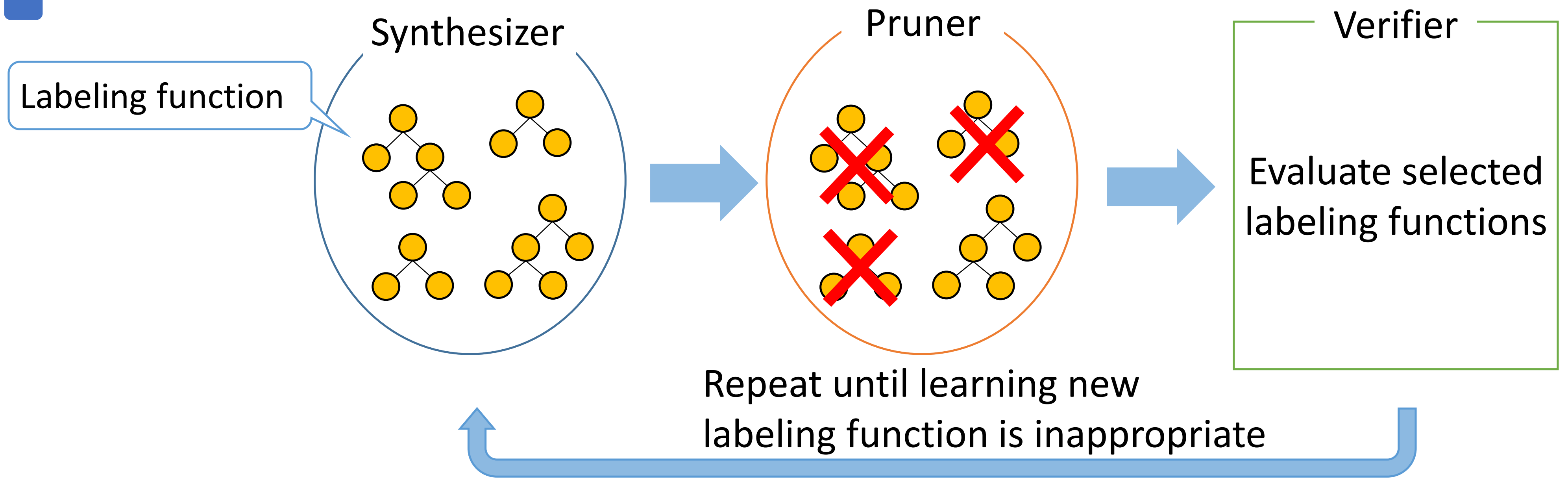
- An automated system that
  - takes <u>a small labeled</u> and <u>a large unlabeled</u> dataset as input
  - outputs <u>probabilistic training labels</u> for the unlabeled data



Label the unlabeled data

Automatically generates labeling functions from labeled data

Snuba



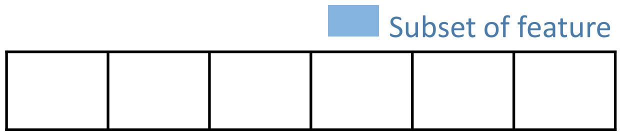
- **Synthesizer**: generates a number of labeling functions from given dataset
- **Pruner**: selects good labeling functions
- **Verifier**: evaluates selected labeling functions and decide whether to stop generating labeling functions or not
  - If we repeat the steps, decide a subset of dataset to generate labeling functions and send to synthesizer

Synthesizer

Labeled data

$$(x_1^L, y_1^L), (x_2^L, y_2^L), \ldots, (x_{N_L}^L, y_{N_L}^L)$$

Feature vector



Generate labeling functions

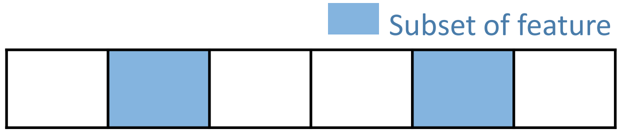
Possible labeling functions :
- Decision Tree
- Logistic Regressor
- K-Nearest Neighbor

Synthesizer

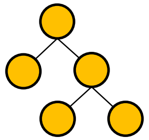
Labeled data

$$(x_1^L, y_1^L), (x_2^L, y_2^L), \ldots, \left(x_{N_L}^L, y_{N_L}^L\right)$$

Feature vector



Generate labeling functions



Generate a labeling function on a subset of features

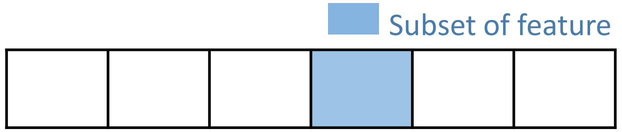
Possible labeling functions :
- Decision Tree
- Logistic Regressor
- K-Nearest Neighbor

Synthesizer

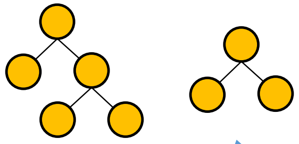
Labeled data

$$(x_1^L, y_1^L), (x_2^L, y_2^L), \dots, (x_{N_L}^L, y_{N_L}^L)$$

Feature vector



Generate labeling functions



Generate a labeling function on a subset of features

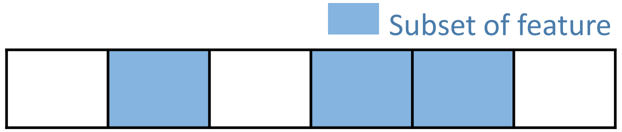
Possible labeling functions :
- Decision Tree
- Logistic Regressor
- K-Nearest Neighbor

Synthesizer

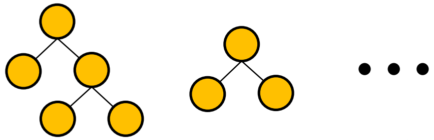
Labeled data

$$(x_1^L, y_1^L), (x_2^L, y_2^L), \ldots, (x_{N_L}^L, y_{N_L}^L)$$

Feature vector



Generate labeling functions



Possible labeling functions :
- Decision Tree
- Logistic Regressor
- K-Nearest Neighbor

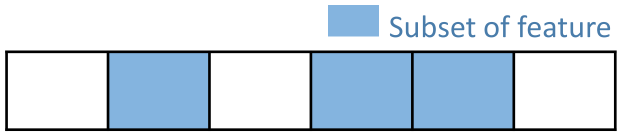
Generate labeling functions for all combinations

Synthesizer

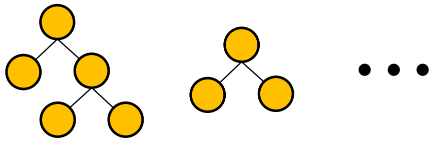
Labeled data

$(x_1^L, y_1^L), (x_2^L, y_2^L), \ldots, (x_{N_L}^L, y_{N_L}^L)$

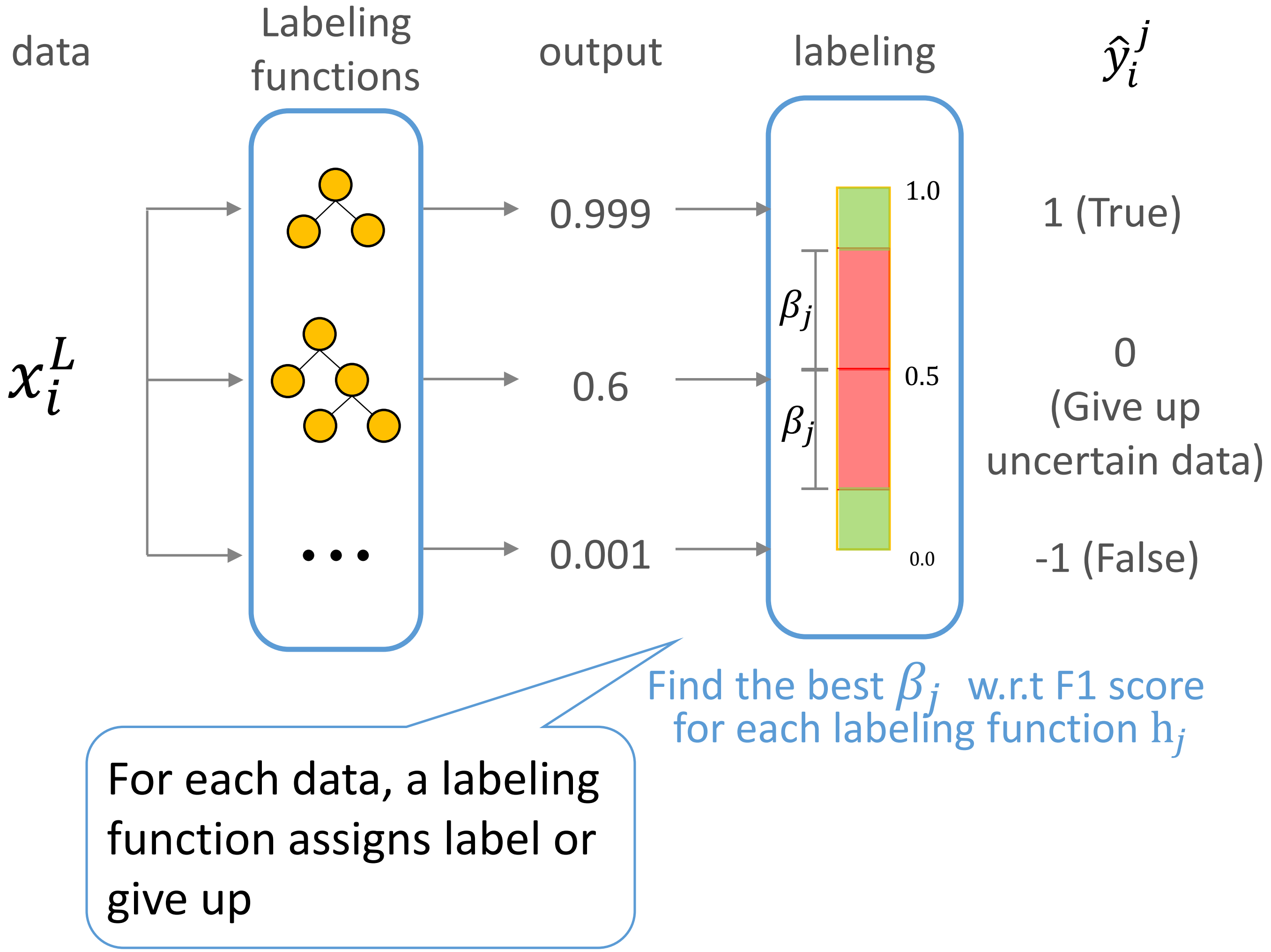
Feature vector



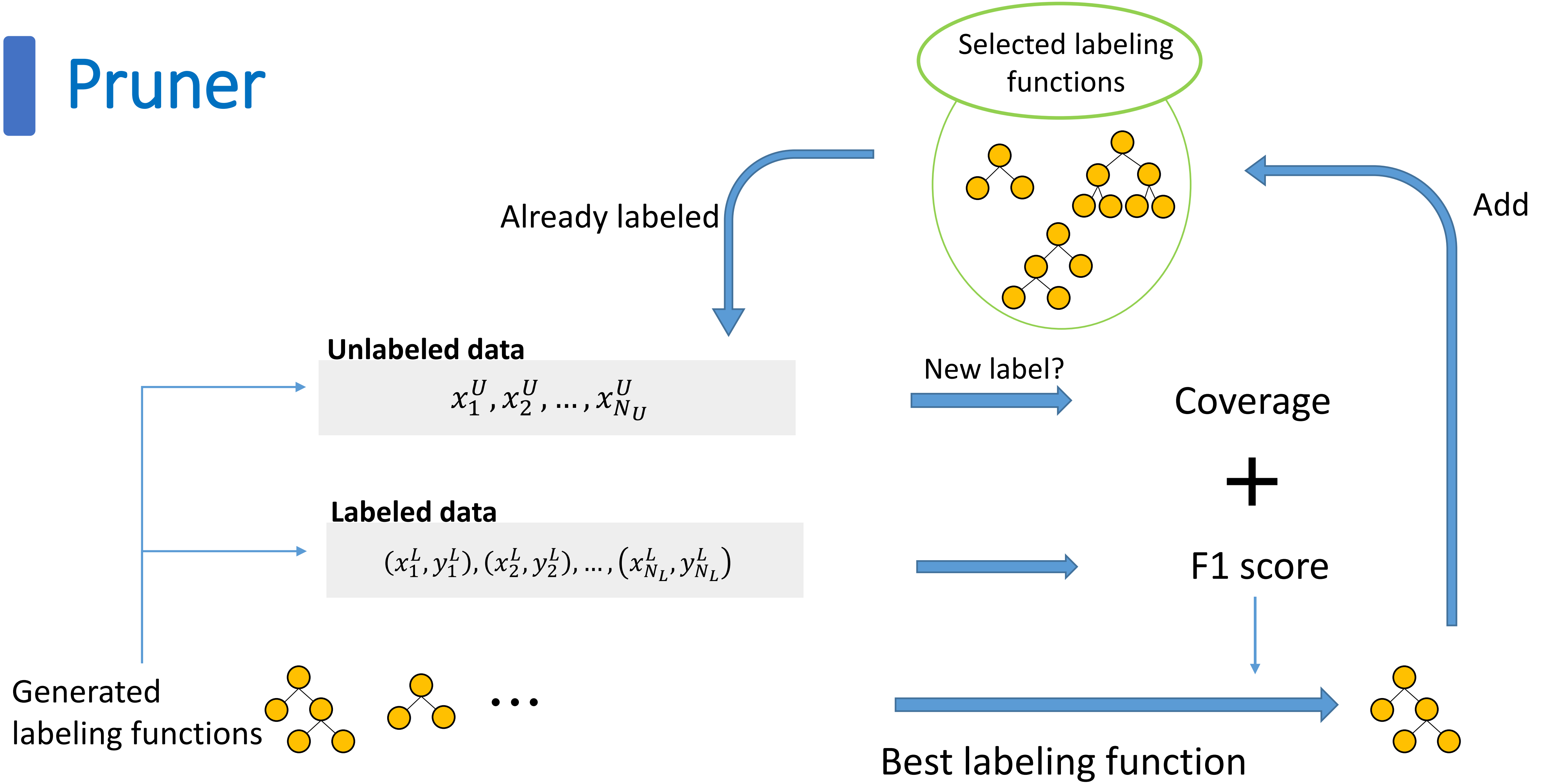
Generate labeling functions



Possible labeling functions :
- Decision Tree
- Logistic Regressor
- K-Nearest Neighbor

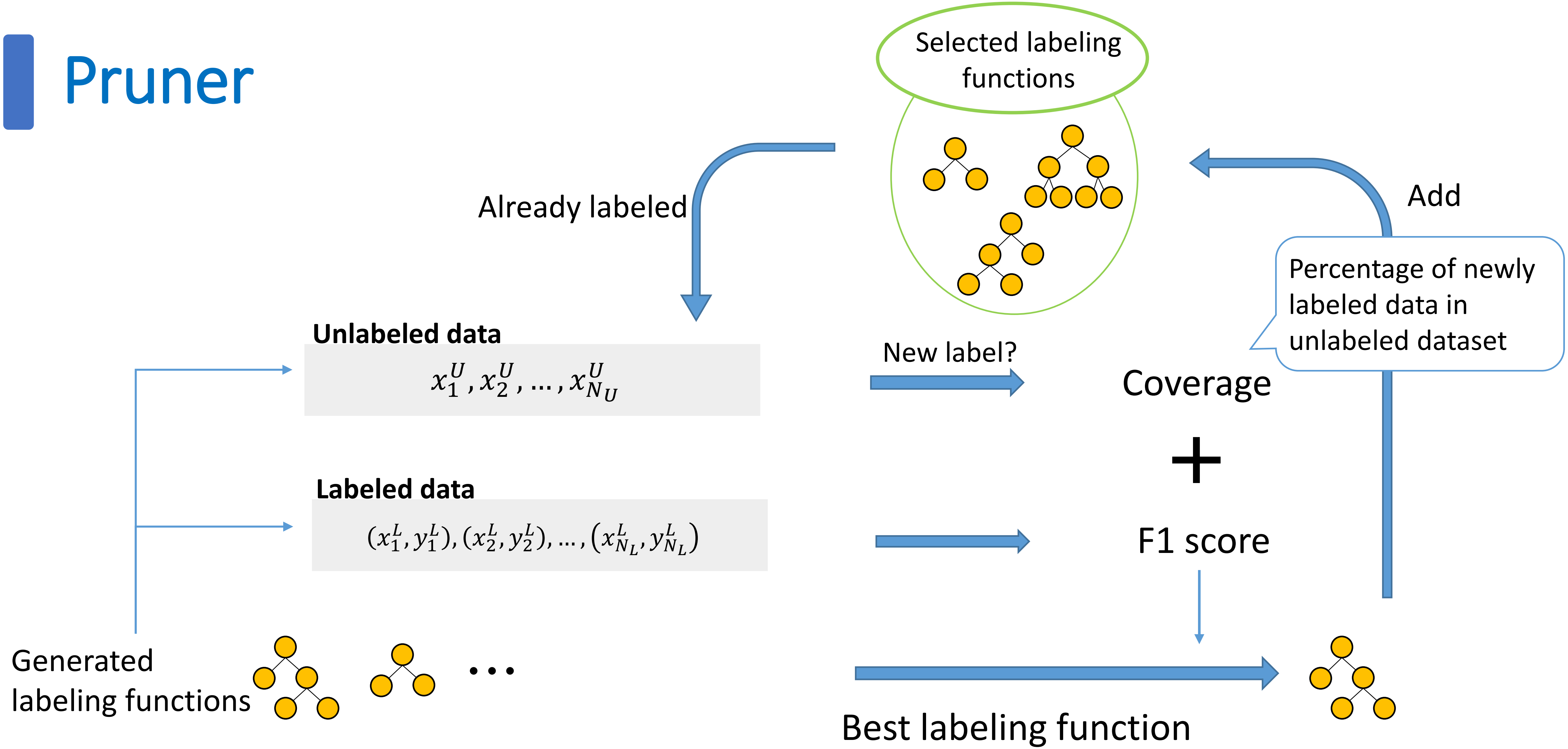


Pruner

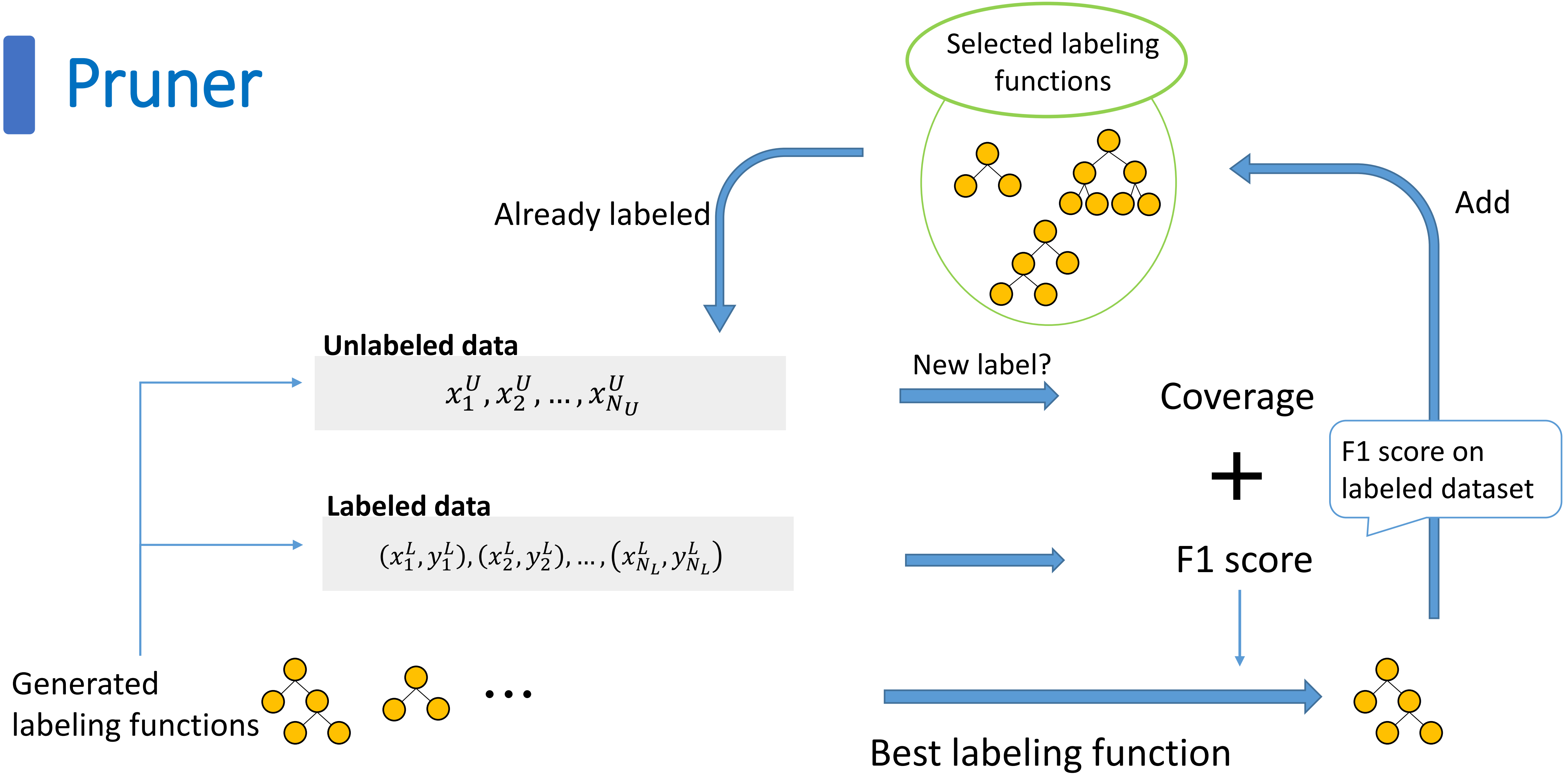


Pruner selects the best labeling function from generated ones

Pruner

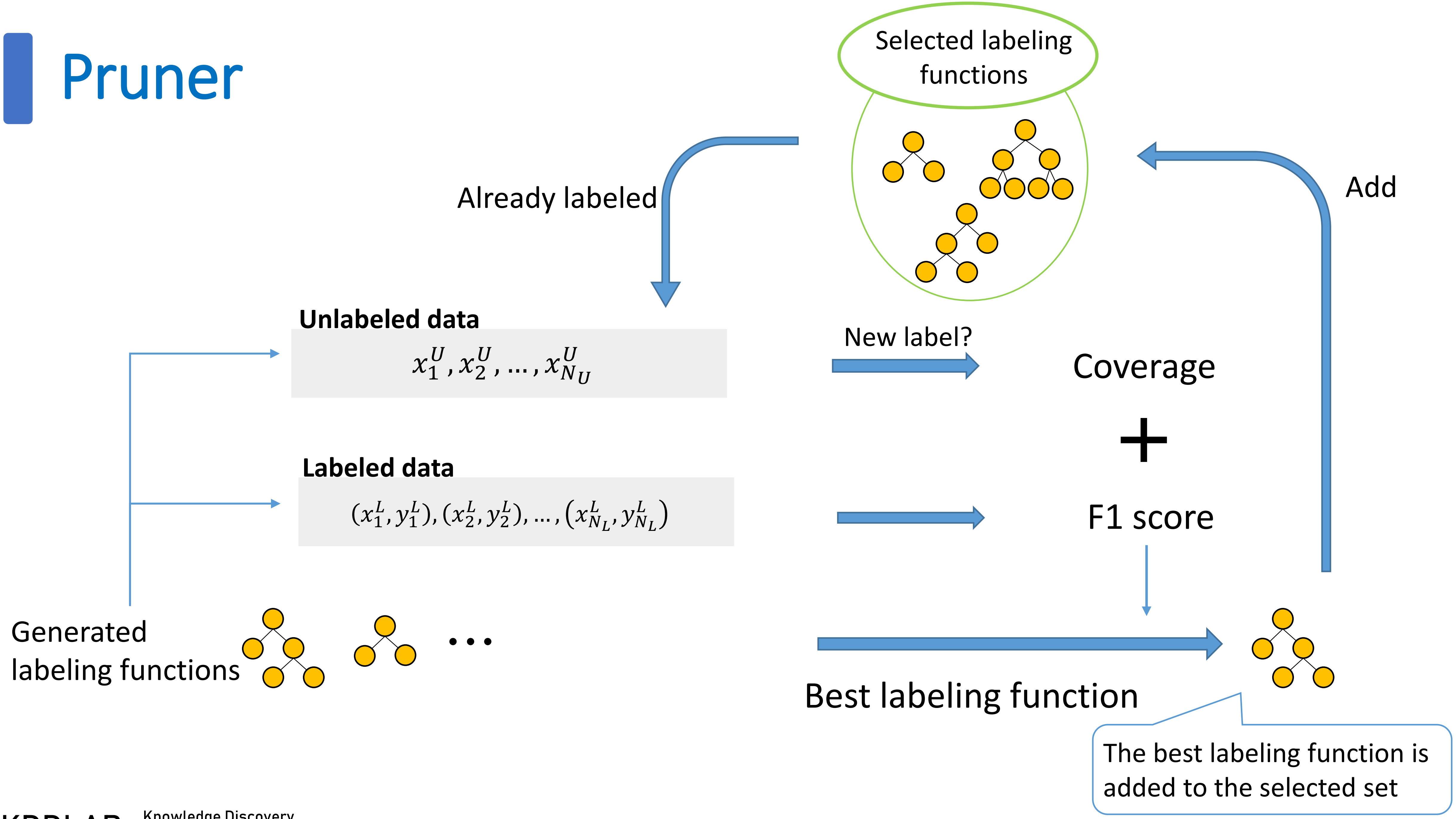

Selected labeling functions
Already labeled
Add
Percentage of newly labeled data in unlabeled dataset
Unlabeled data
$x_1^U, x_2^U, \dots, x_{N_U}^U$
New label?
Coverage
+
Labeled data
$(x_1^L, y_1^L), (x_2^L, y_2^L), \dots, (x_{N_L}^L, y_{N_L}^L)$
F1 score
Generated labeling functions
...
Best labeling function

Pruner

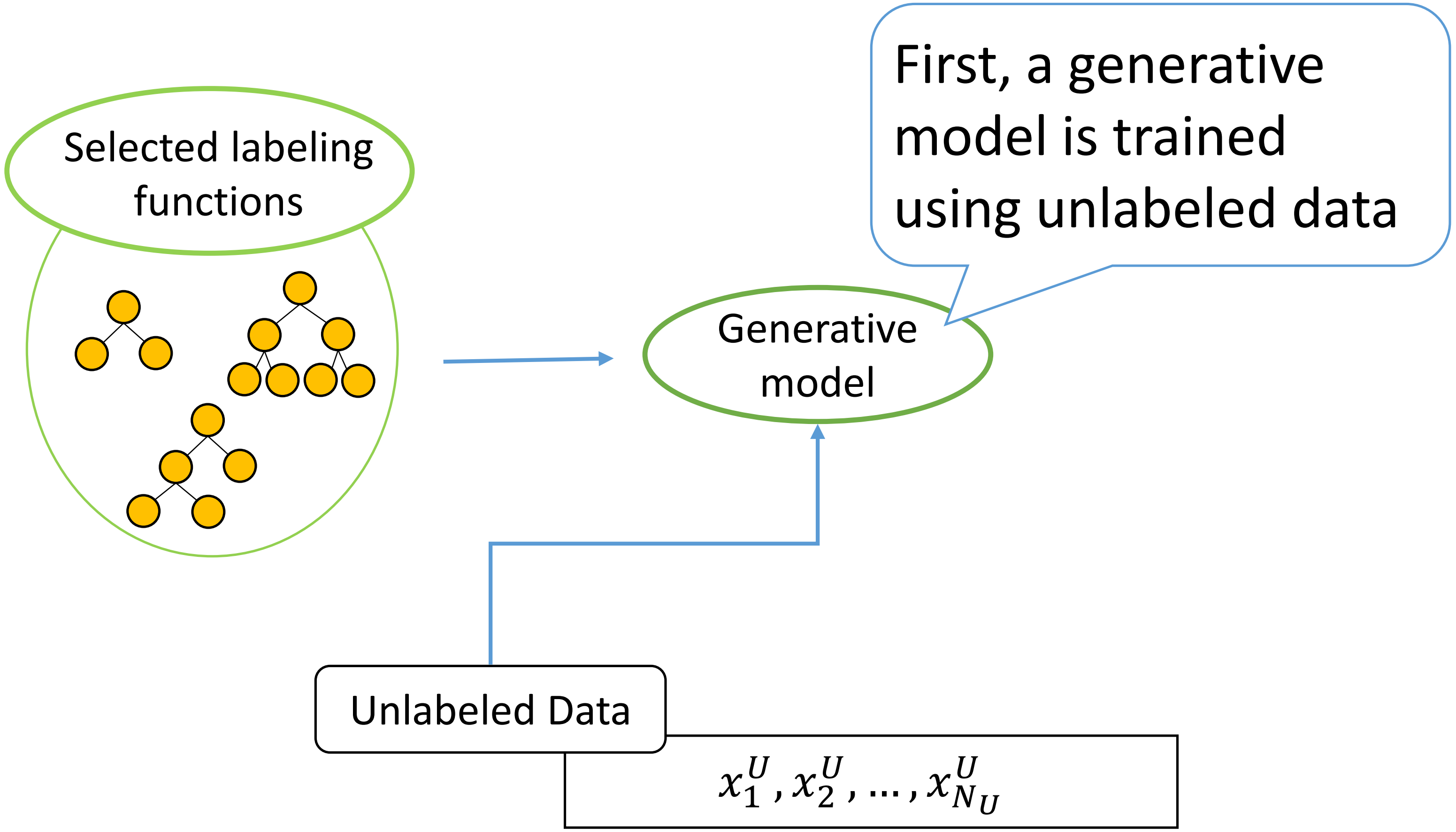


Selected labeling functions
Already labeled
Add
Unlabeled data
$x_1^U, x_2^U, \dots, x_{N_U}^U$
New label?
Coverage
+
F1 score on labeled dataset
Labeled data
$(x_1^L, y_1^L), (x_2^L, y_2^L), \dots, (x_{N_L}^L, y_{N_L}^L)$
F1 score
Generated labeling functions
…
Best labeling function

Pruner

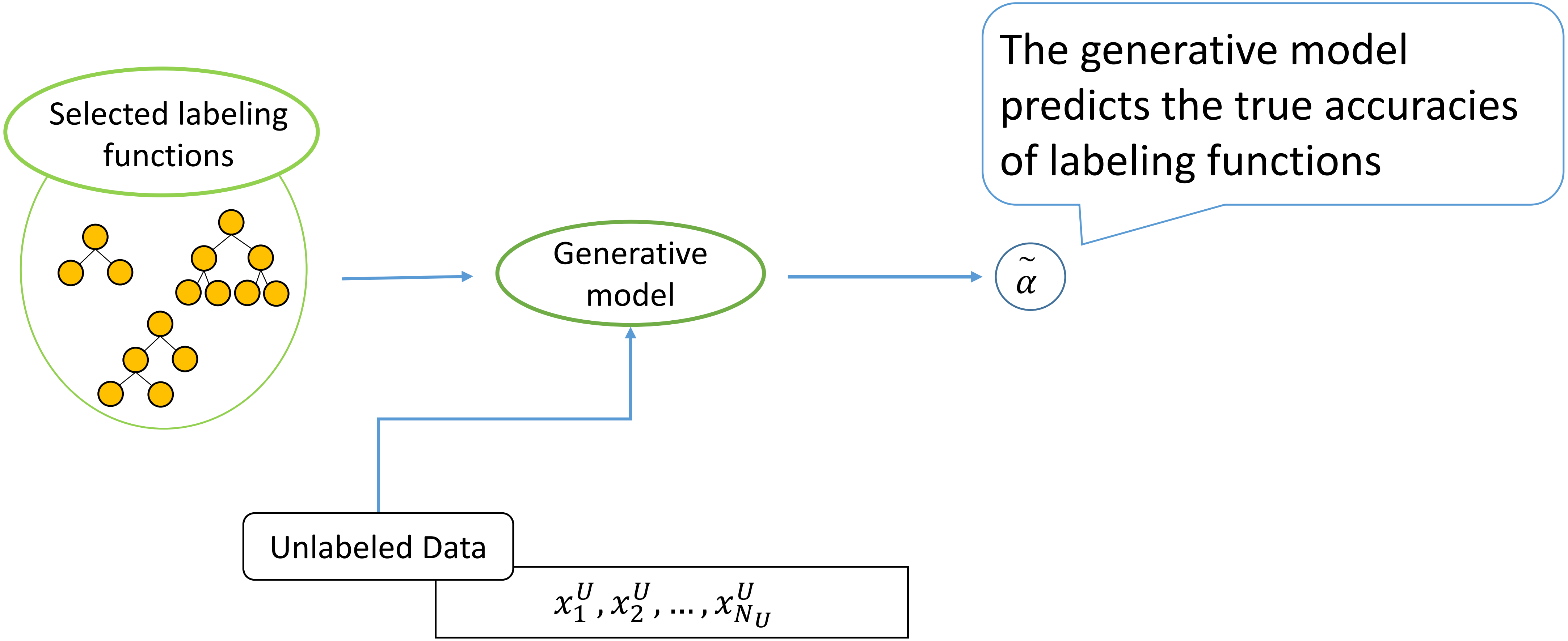

Selected labeling functions
Already labeled
Add
Unlabeled data
$x_1^U, x_2^U, \ldots, x_{N_U}^U$
New label?
Coverage
+
Labeled data
$(x_1^L, y_1^L), (x_2^L, y_2^L), \ldots, (x_{N_L}^L, y_{N_L}^L)$
F1 score
Generated labeling functions
…
Best labeling function
The best labeling function is added to the selected set

Verifier



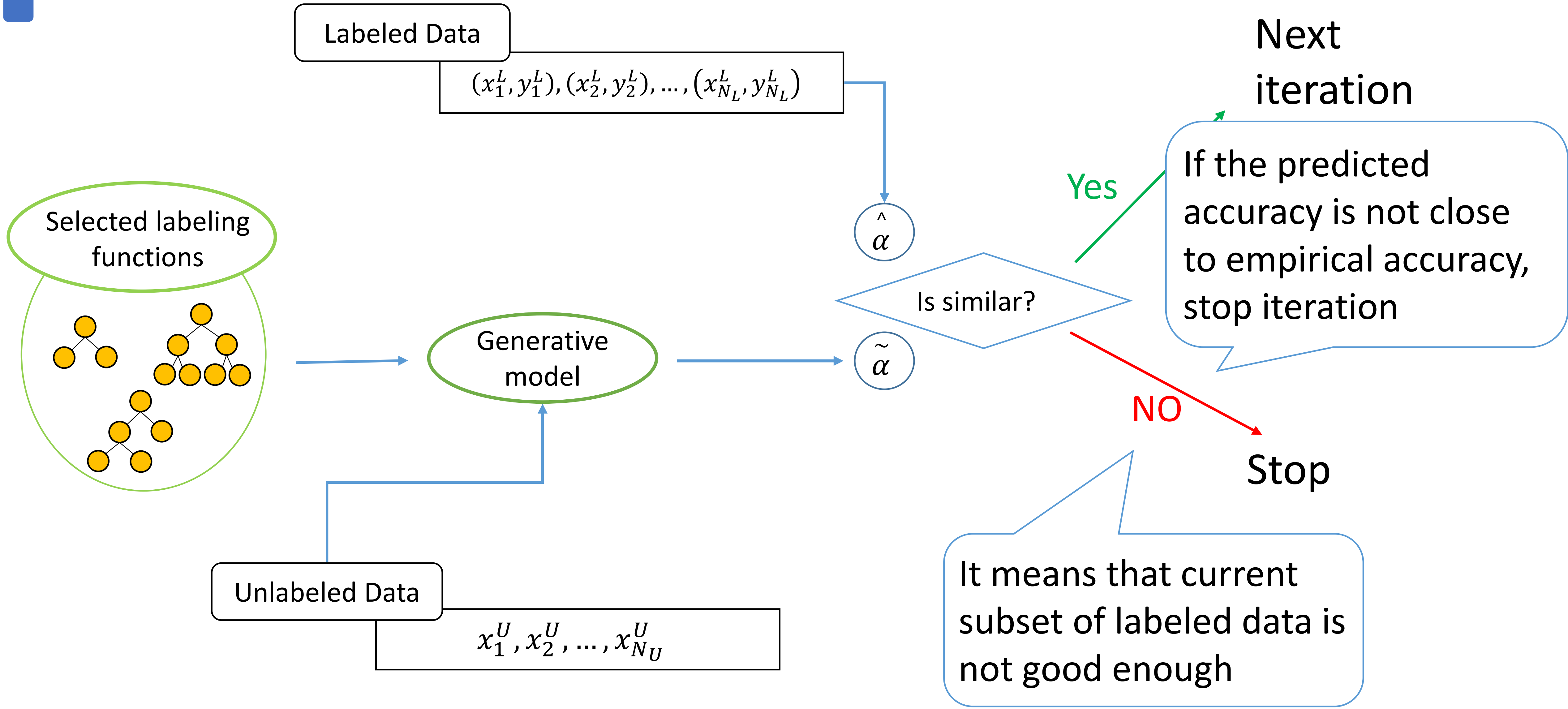
Selected labeling functions
Generative model
First, a generative model is trained using unlabeled data
Unlabeled Data
$x_1^U, x_2^U, \ldots, x_{N_U}^U$

Verifier



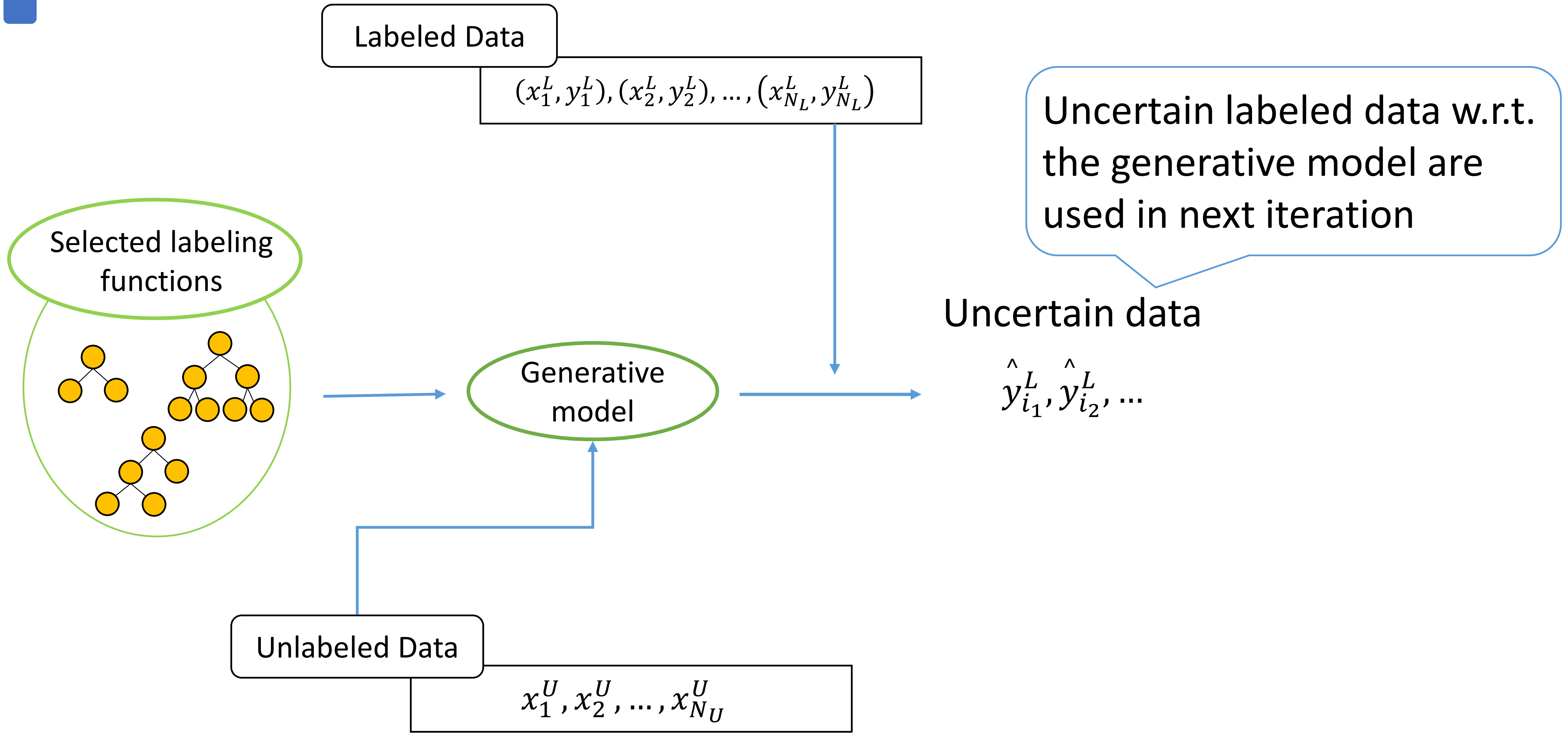
Selected labeling functions
Generative model
$\tilde{\alpha}$
The generative model predicts the true accuracies of labeling functions
Unlabeled Data
$x_1^U, x_2^U, \ldots, x_{N_U}^U$

Verifier



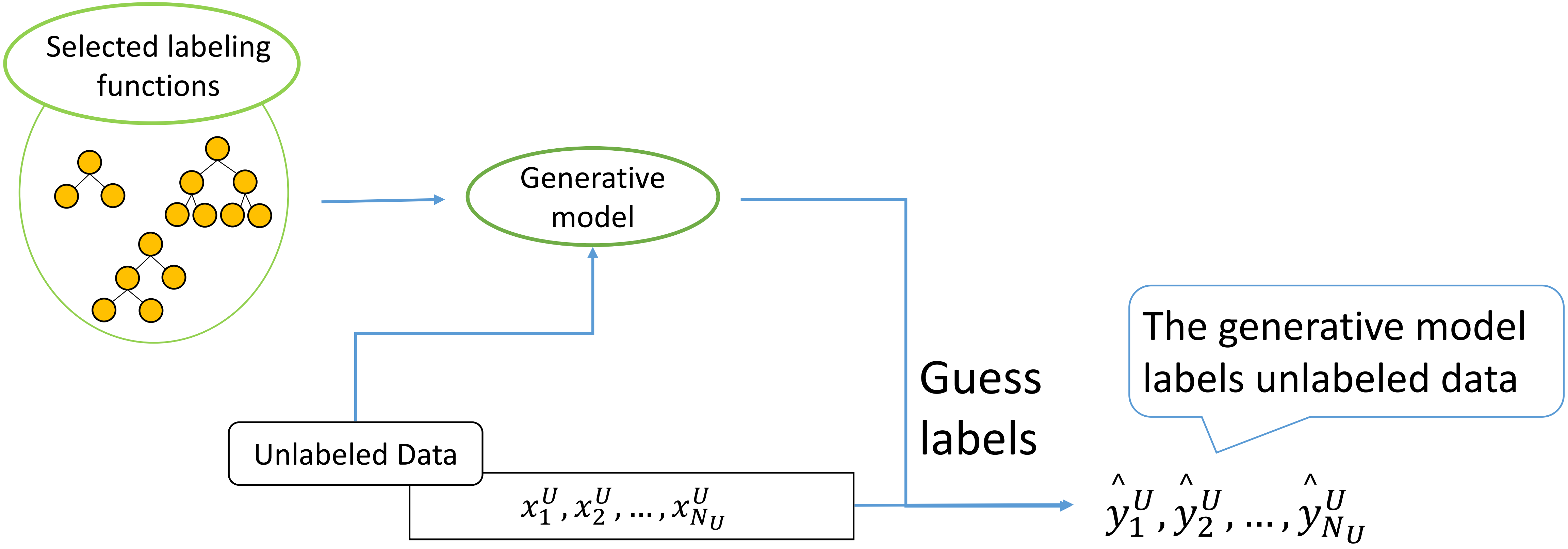
Labeled Data
$(x_1^L, y_1^L), (x_2^L, y_2^L), \dots, (x_{N_L}^L, y_{N_L}^L)$
Selected labeling functions
Generative model
Unlabeled Data
$x_1^U, x_2^U, \dots, x_{N_U}^U$
$\hat{\alpha}$
$\tilde{\alpha}$
Is similar?
Yes
Next iteration
If the predicted accuracy is not close to empirical accuracy, stop iteration
NO
Stop
It means that current subset of labeled data is not good enough

Verifier



Labeled Data
$(x_1^L, y_1^L), (x_2^L, y_2^L), \dots, (x_{N_L}^L, y_{N_L}^L)$
Uncertain labeled data w.r.t. the generative model are used in next iteration
Selected labeling functions
Generative model
Uncertain data
$\hat{y}_{i_1}^L, \hat{y}_{i_2}^L, \dots$
Unlabeled Data
$x_1^U, x_2^U, \dots, x_{N_U}^U$

Verifier



Selected labeling functions
Generative model
Guess labels
The generative model labels unlabeled data
Unlabeled Data
$x_1^U, x_2^U, \ldots, x_{N_U}^U$
$\hat{y}_1^U, \hat{y}_2^U, \ldots, \hat{y}_{N_U}^U$

Experiments

- Datasets
  - Image classification
    - **Bone Tumor**: Tumor classification
    - **Mammogram**: Tumor classification
    - **Visual Genome**: Identifying person on bike
  - Text and Multi-Modal classification
    - **MS-COCO**: Object detection
    - **IMDB**: Plot summary classification
    - **Twitter**: Sentiment analysis
    - **CDR**: Chemical-Disease relation extraction
    - **Hardware**: Classifying valid specification of hardwares

Experiments

- Baselines
  - Decision Tree
  - Boosting (AdaBoost)
    - Adjust weights in Random Forest
  - Transfer Learning
    - Tune the last layer of a pre-trained model
  - Semi-Supervised Learning [NIPS 2004]
    - Propagates labels to nearby points
  - UDF (User-Driven labeling Functions, by **Snorkel**)

Experiments

		Snuba Improvement Over				
Application	**Snuba F1 Score**	**Decision Tree**	**Boosting**	**Transfer Learning**	**Semi-Supervised**	**UDF**
Bone Tumor	71.55	+6.37	+8.65	-	+6.77	+9.13
Mammogram	74.54	+5.33	+5.02	+5.74	+3.26	+9.74
Visual Genome	56.83	+7.62	+6.20	+5.58	+5.94	+6.38
MS-COCO	69.52	+1.65	+2.70	+2.51	+1.84	+2.79
IMDb	62.47	+7.78	+12.12	+3.36	+14.35	+3.67
Twitter	78.84	+5.03	+4.43	-	+3.84	+13.8
CDR	41.56	+5.65	+11.22	-	+7.49	-12.24
Hardware	68.47	+5.20	+4.16	-	+2.71	-4.75*

- Report of Snuba is the **result of an end model** trained on labels generated by Snuba
- Snuba **outperforms** other methods, except UDF on CDR and Hardware

Experiments

Application	User Heuristics F1	P	R	Snuba Heuristics F1	P	R	Lift(F1)
Bone Tumor	30.91	89.47	18.68	31.58	33.75	29.67	+0.67
Visual Genome	34.76	98.28	21.11	46.06	48.10	44.19	+11.30
MS-COCO	21.43	63.66	12.88	24.41	29.40	41.49	+12.98
IMDb	20.65	76.19	11.94	46.47	48.03	45.52	+25.82

- Labeling functions generated by Snuba (Snuba Heuristics) show lower precision but **higher recall** than user heuristics

Experiments

Application	Snuba Heuristics F1	Snuba Heuristics P	Snuba Heuristics R	Snuba + End Model F1	Snuba + End Model P	Snuba + End Model R	Lift(F1)
Bone Tumor	31.58	33.75	29.67	71.55	58.86	91.21	+39.97
Visual Genome	46.06	48.10	44.19	56.83	41.34	90.91	+10.77
MS-COCO	24.41	29.40	41.49	69.52	55.80	92.16	+35.11
IMDb	46.47	48.03	45.52	62.47	45.42	100.	+16.00

- **Discriminative model** on **labels generated by Snuba** highly **improves** both precision and recall

GOGGLES: Automatic Image Labeling with Affinity Coding

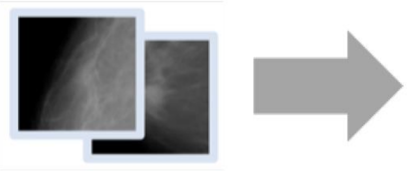
Nilaksh Das, Sanya Chaba, Renzhi Wu, Sakshi Gandhi, Duen Horng Chau, Xu Chu

SIGMOD 2020

LFs (Labeling Functions) for Image Labeling

- Existing works (Snorkel, Snuba) require associated **metadata** for each image
  - Text annotations (e.g., medical notes associated with X-Ray images)
  - Primitives (e.g., bounding boxes for X-Ray images)
- The metadata is **difficult** to obtain
  - In Snuba, **radiologists** have pre-extracted bounding boxes for X-Ray images
- For every **new dataset,** a **new set** of labeling functions is required
- In this paper, the authors propose a method to automatically generate probabilistic labels for images without such metadata

Tumor
(benign vs. malignant)



```
def labeling_function_1(x)
  Obtain primitive bounding_box from x
  If bounding_box.area > 210.8:
    return False
  If bounding_box.area < 150:
    return Abstain
```

Overview of GOGGLES

- Step 1: Similarity Matrix Construction
  - Similarity matrix consists of similarity scores between all pairs of samples w.r.t. multiple similarity functions

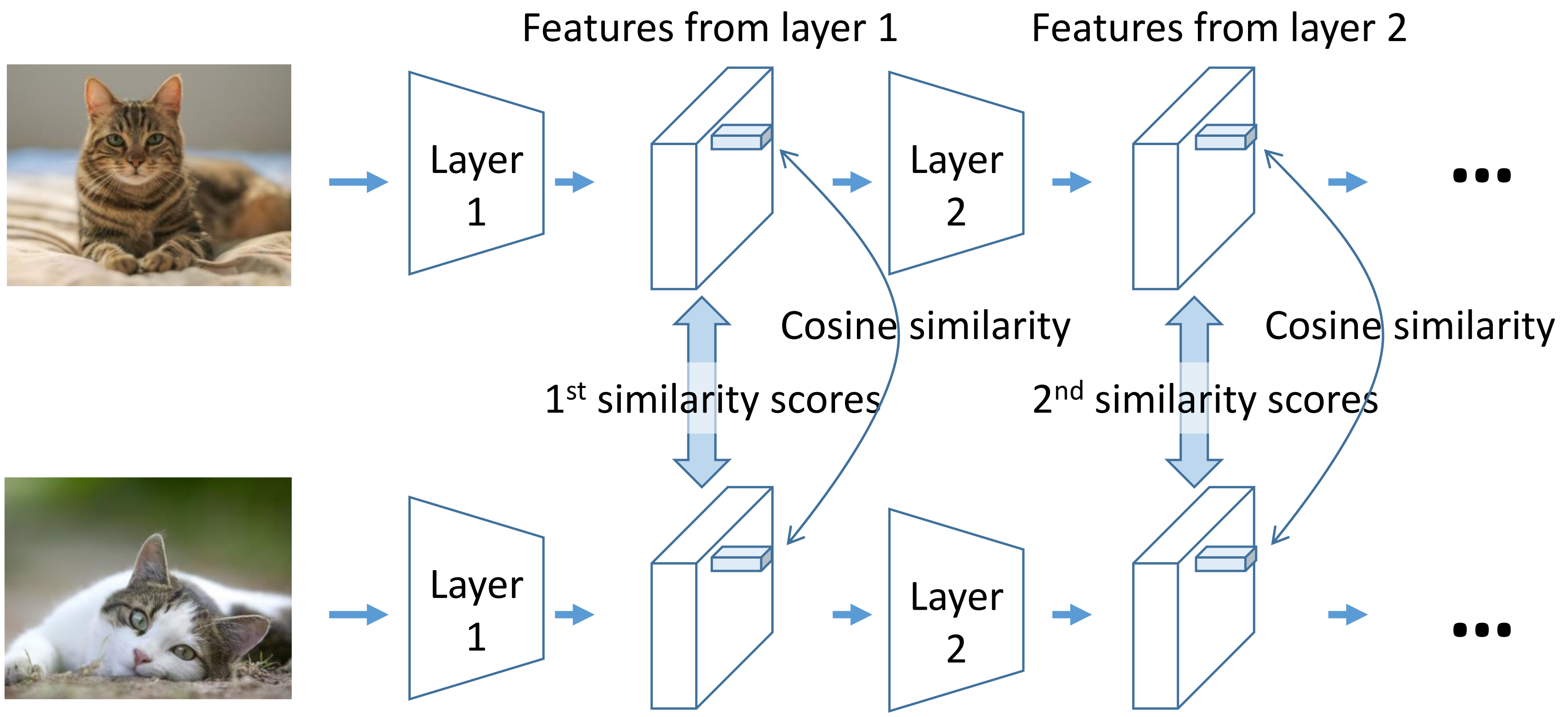
- Step 2: Class Inference
  - Cluster unlabeled and labeled data together using the similarity matrix
  - Find cluster-to-class mapping with a small number of labeled data

Problem Definition

- Given
  - A large number of unlabeled data
  - A small number of labeled data
  - K: the number of classes
- Find
  - The **probabilistic labels** of the unlabeled data

Computing Similarity Scores

- Pre-trained convolutional neural networks (CNN) is utilized
- Between the features extracted from each layer of CNN, similarity scores are obtained
  - Similarity scores are computed based on cosine similarity between vectors

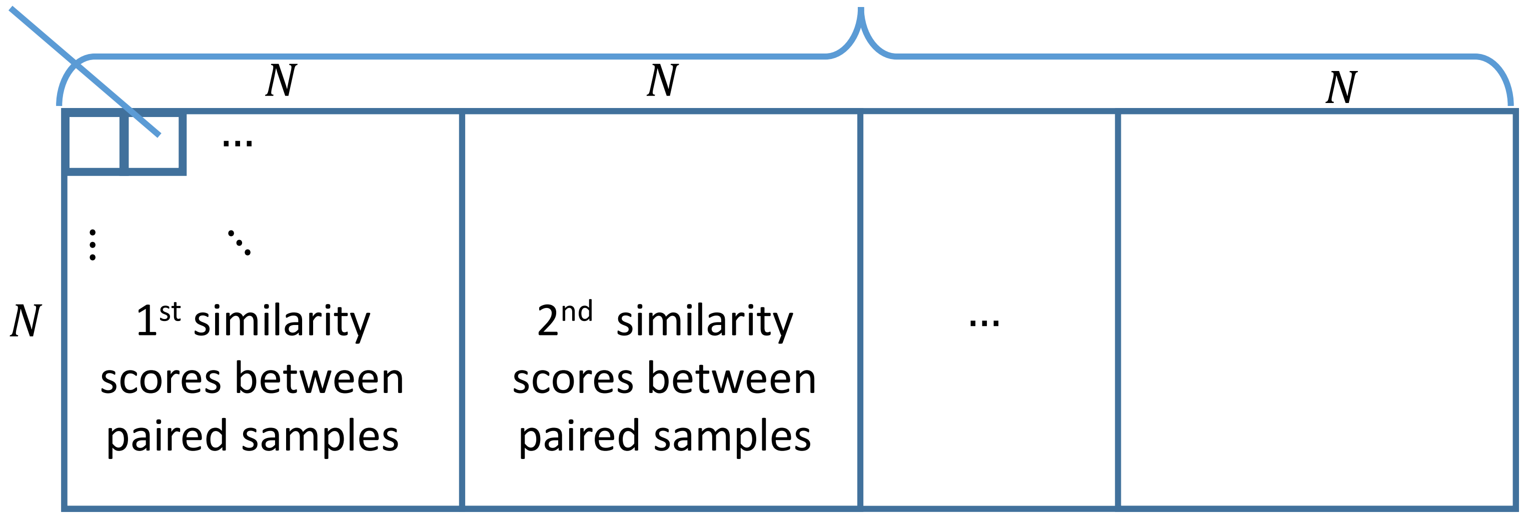


Similarity Matrix Construction

- α: the number of similarity scores between a pair of samples
- N: the number of all examples
- $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N \times \alpha N}$: the similarity matrix

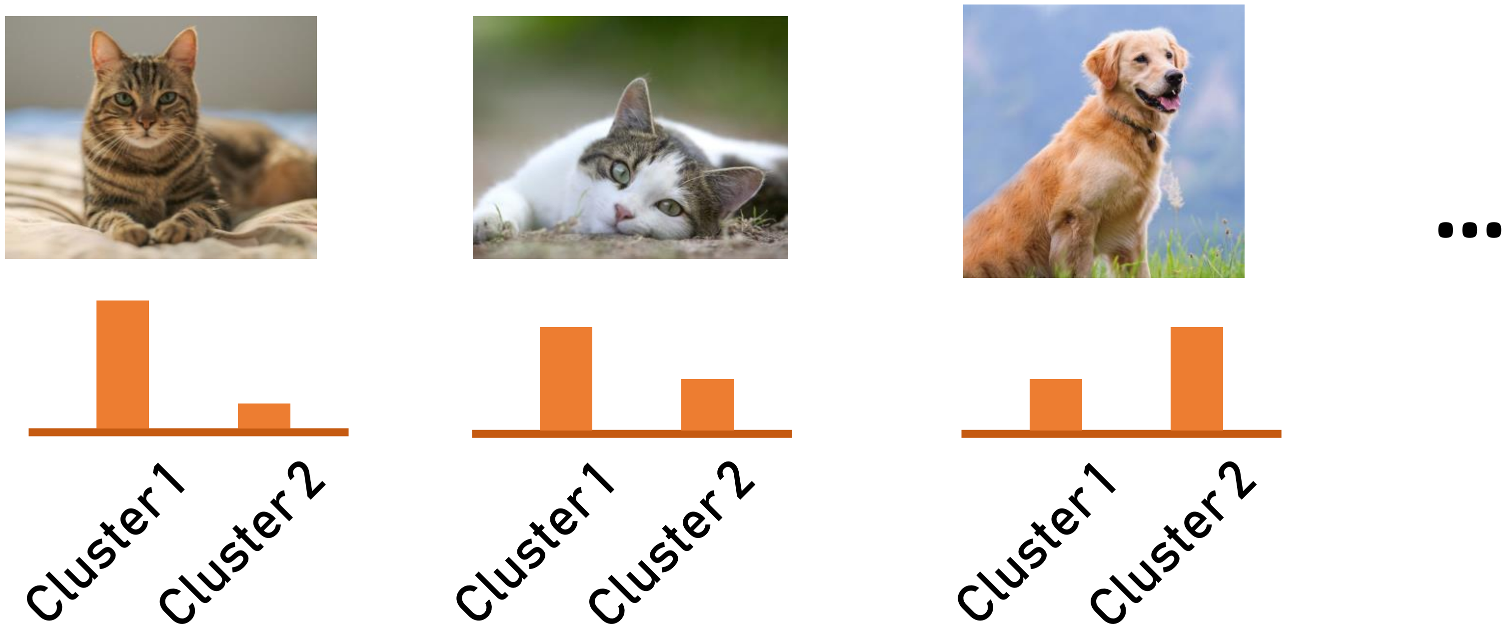
x_i: the i-th sample

The 1st similarity score between x_1 and x_2 sample



Clustering based on the Similarity Matrix

- The i-th row of the similarity matrix $\boldsymbol{\mathcal{A}}$ is considered as the feature vector of the i-th sample
- Apply the GMM (Gaussian mixture model) to the feature vectors from $\boldsymbol{\mathcal{A}}$ where the number of clusters is equal to the number of classes K
- We obtain (soft) cluster assignments of data



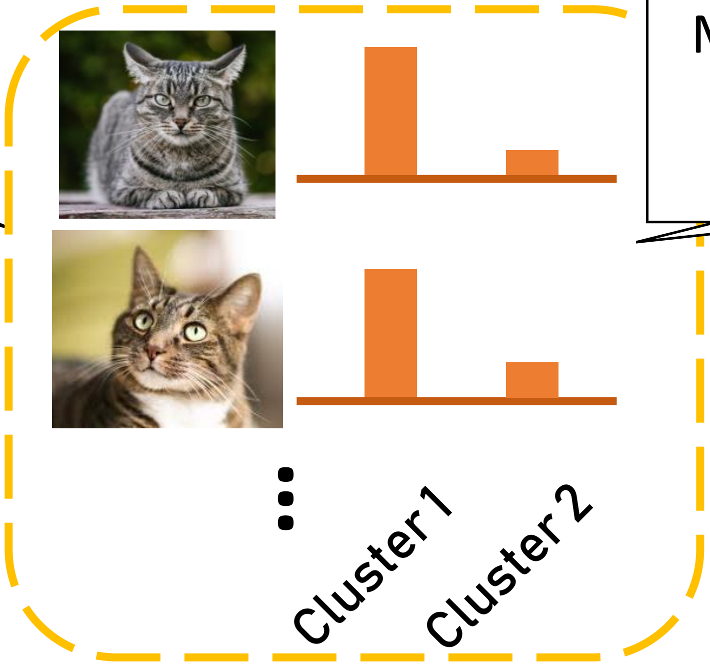
Finding Cluster-to-class Mapping

- Utilize the cluster assignment results for the labeled data
- Find the cluster-to-class mapping $g(k)$ which maximize the sum of assignment probability

$$\sum_{k=1}^{K} \sum_{i \in LS_{g(k)}} P(y_i = k)$$

  - $LS_{k'}$: the set of the indices of labeled examples for class k'

The labeled data for cat



Most data are clustered as Cluster 1

Cluster 1 → Cat

Datasets

- **CUB**: bird species classification
  - Provides image-level attribute annotations that help explain the visual characteristics of the bird in the image, e.g., white head, grey wing etc.
- **GTSRB**: traffic sign classification
- **Surface**: metallic surface classification
- **TB-Xray**: classification for normal lung X-ray and abnormal X-ray
- **PN-Xray**: pneumonia chest X-ray classification

Compared Methods

- **Snorkel** [Ratner, Bach, Ehrenberg, Fries, Wu and Ré: PVLDB 2017]
- **Snuba** [Varma and Ré: PVLDB 2018]: utilizing a pretrained CNN network to extract feature
- **FSL** [Chen, Liu, Kira, Wang and Huang: ICLR 2019]: a domain adaptation method using pretrained CNN network and a small number of labeled data

Evaluation of Labeling Accuracy

- GOGGLES outperforms the other data programming methods in terms of labeling accuracy

Dataset	GOGGLES (our results)	Data Programming	
		Snorkel	Snuba
CUB	97.83	89.17	58.83
GTSRB	70.51	-	62.74
Surface	89.18	-	57.86
TB-Xray	76.89	-	59.47
PN-Xray	74.39	-	55.50
Average	**81.76**	-	58.88

CUB is the **only dataset** having **metadata** to design labeling functions

Comparison of Discriminator Accuracy

- The generated probabilistic labels are used to train the downstream classification model
  - A convolutional neural network (VGG-16) is used for the discriminator
- GOGGLES outperforms the other methods

Dataset	FSL	Snorkel	Snuba	GOGGLES
CUB	84.74	87.85	56.32	95.30
GTSRB	90.72	-	70.11	91.54
Surface	76.00	-	51.67	83.33
TB-Xray	66.42	-	62.71	70.90
PN-Xray	68.28	-	62.19	69.06
Average	77.23	-	60.60	**82.03**

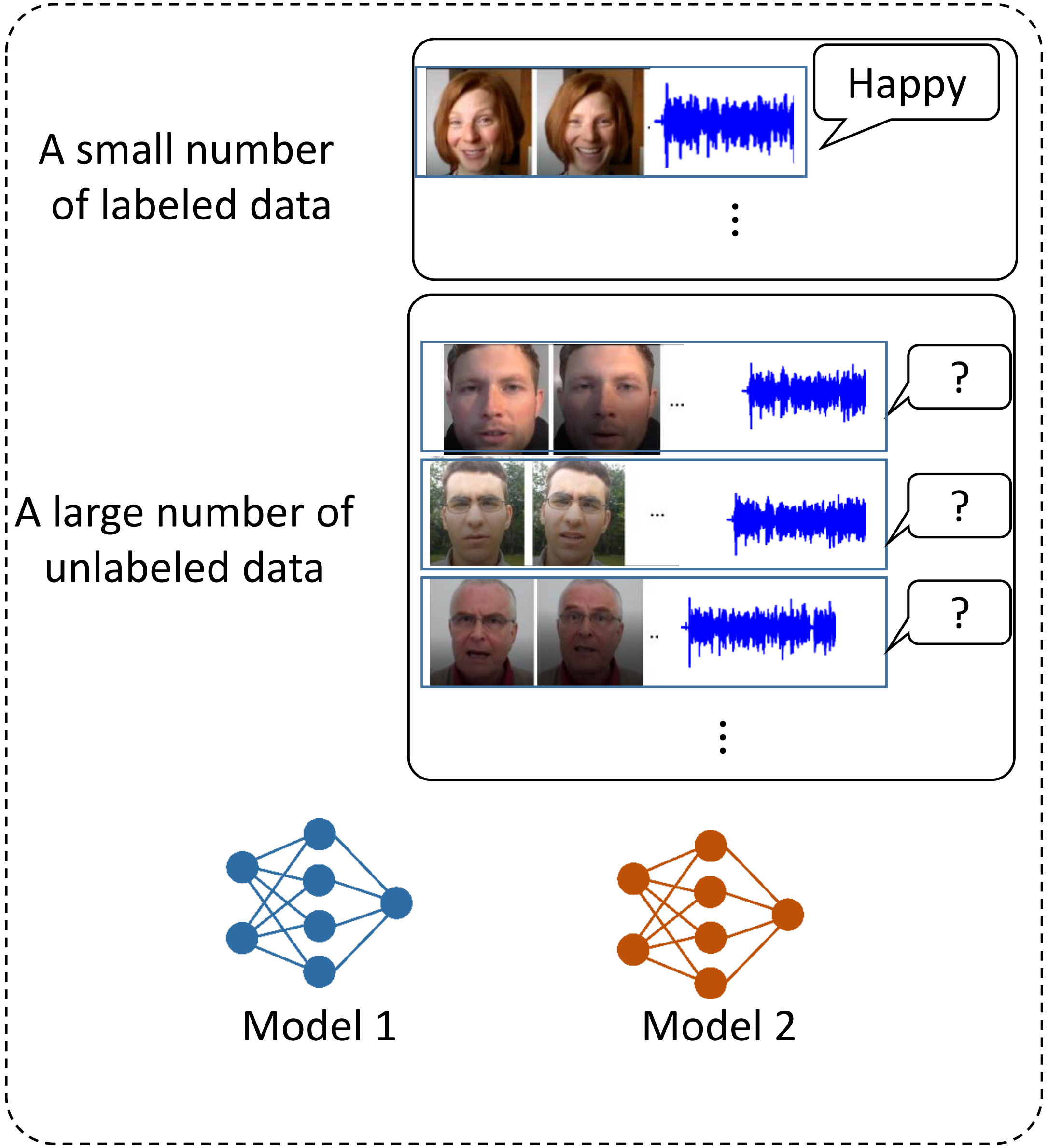
Self-paced Multi-view Co-training

Fan Ma, Deyu Meng, Xuanyi Dong, Yi Yang

JMLR 2020

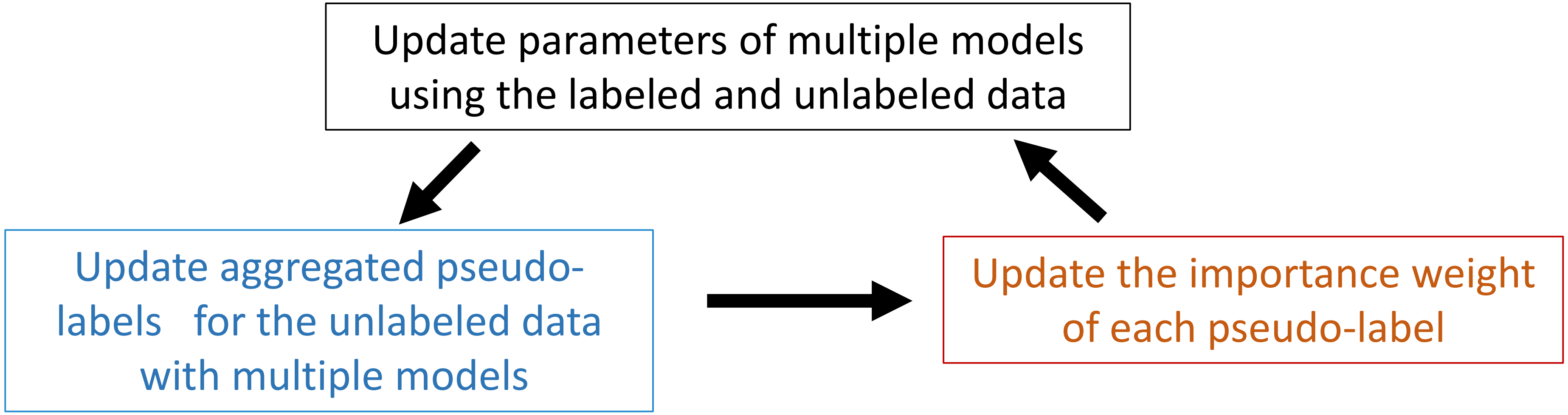
Self-paced Multi-view Co-training (SPamCo)

- Aggregate multiple models' outputs to generate pseudo labels and update models using the pseudo labels
- Problem definition
  - Given
    - A small number of labeled data
    - A large number of unlabeled data
    - Multiple classifiers on different modalities
  - Find
    - The labels of unlabeled data



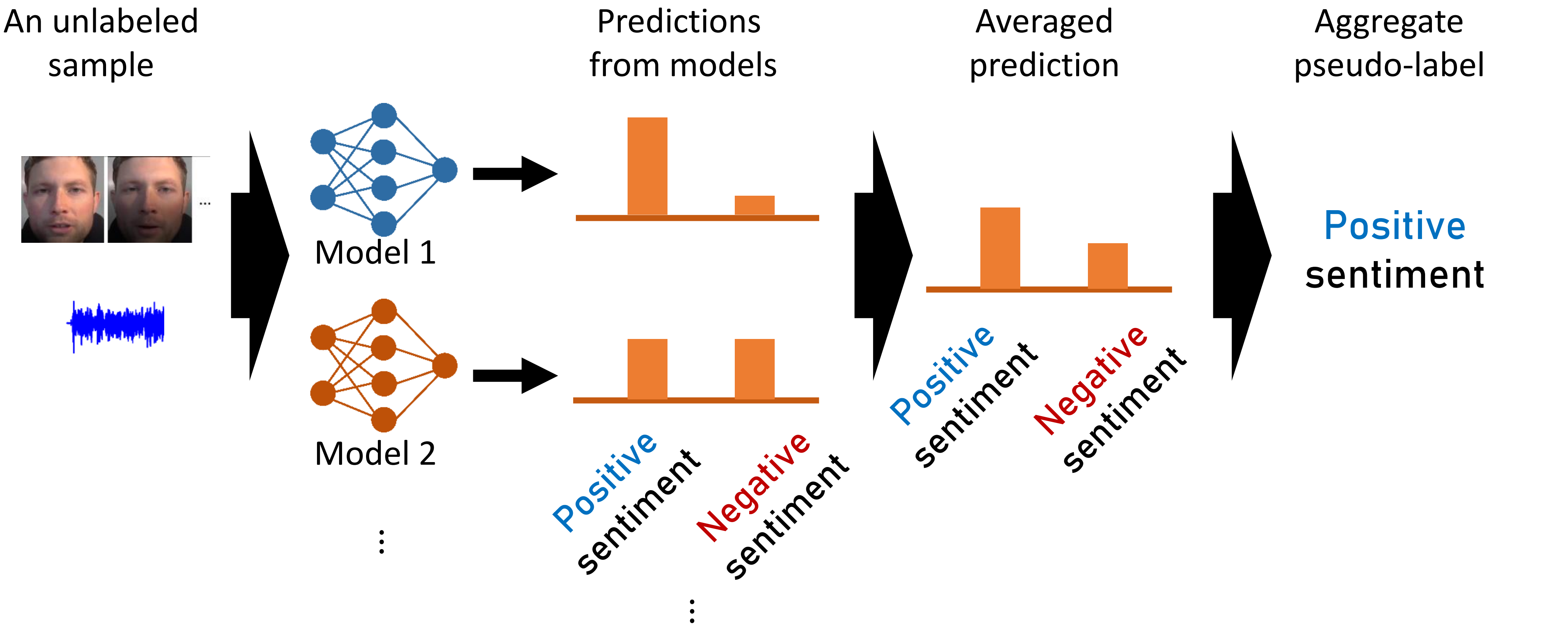
Overview of SPamCo

- **Initially**, each classifier is **trained with the labeled data**
- **Repeat** the following steps
  - Find the **aggregated pseudo-labels** for the unlabeled data with the multiple models
  - Compute the **importance weight** of **each pseudo-label** for training
  - **Train each classifier** by using the labeled data and the unlabeled data with aggregated pseudo-labels and importance weights



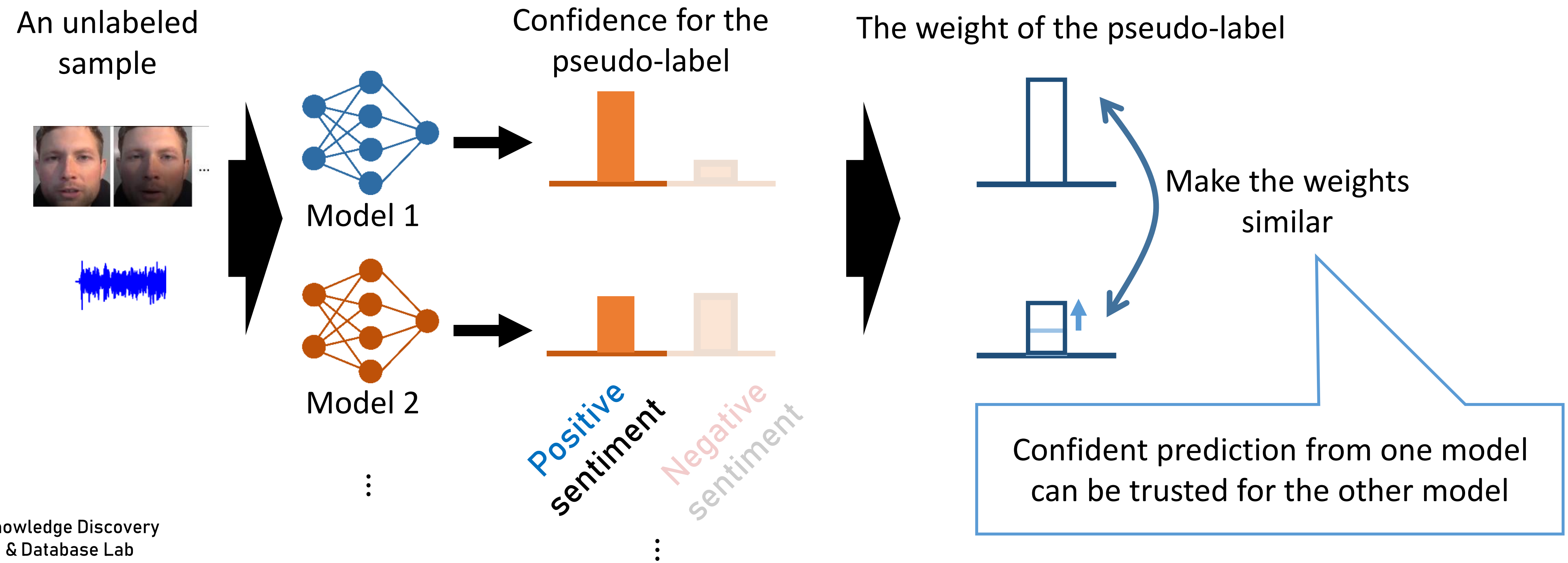
Generating Aggregated Pseudo-labels

- The pseudo-label of each sample is generated by using the averaged prediction from models



The Importance Weight of a Pseudo-label

- For each model, each pseudo-label's importance weight for training is computed
  - **The smaller the confidence** for the pseudo-label is, **the smaller the weight** of the pseudo-label becomes
  - There exists a regularization to **make the weights similar** across the models



Experimental Result

- Task: person re-identification on Market-1501 dataset
- Evaluation measure: the area under the Precision-Recall curve
- Base classifiers: Resnet-50 and DenseNet-121

Ensembled by averaging

- **Base**: Use only labeled data
- **SelfTrain**: Each classifier use its own pseudo-label
- **Cotrain**: Pseudo-labels are exchanged with each other

	Resnet-50	DenseNet-121	Ensemble
Base	40.5±1.57	38.5±1.20	47.7±0.78
SelfTrain	59.2±0.70	61.7±1.14	67.7±0.72
Cotrain	59.3±0.50	61.9±0.80	67.0±0.33
Cotrain(Rep)	60.1±0.72	62.5±0.77	67.7±0.42
SPamCo(soft)	**61.7±0.21**	**64.7±0.66**	**69.5±0.33**

KDDLAB
Knowledge Discovery
& Database Lab

Dual Supervision Framework for Relation Extraction with Distant Supervision and Human Annotation

Woohwan Jung, Kyuseok Shim

COLING 2020

Dual Supervision Framework [COLING 2020]

- Relation extraction (RE)
  - Task to identify the semantic relationship between entities from text

[Seoul] is the capital city of **[Korea]**

Relation extraction

capital

  - Training deep RE models requires a huge amount of labeled data in the form of entity-annotated text and corresponding relations

Dual Supervision Framework [COLING 2020]

- Human annotation
  - Accurate
  - Expensive
- Distant supervision [M. Mintz, S. Bills, R. Snow, D. Jurafsky: ACL-IJCNLP 09]
  - Using knowledge base, automatically generate labels
  - Easy to obtain a large-scale data
  - Less accurate

Dual Supervision Framework [COLING 2020]

- There exists a **labeling bias** in distant supervision
- Definition of inflation

$$Inflation(r) = \frac{\Pr[r] \text{ in } \textbf{distantly supervised} \text{ data}}{\Pr[r] \text{ in } \textbf{human annotated} \text{ data}}$$

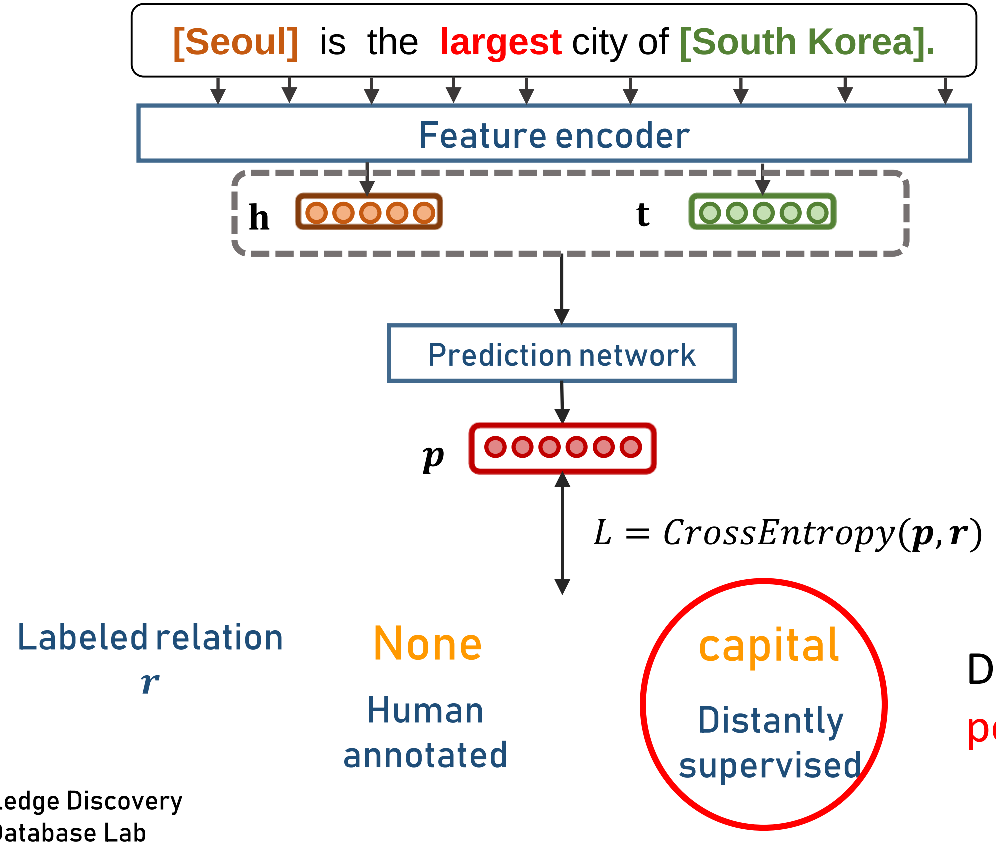
  - $Inflation$(sister_city)=68.03
  - $Inflation$(capital)=12.18
  - …

Distant supervision generates more than 10 times of labels for some relations!

- The labeling bias can degrade the accuracy of models when we use distant supervision in addition to human annotations

Dual Supervision Framework [COLING 2020]

- Existing RE models
  - The human annotated label and distantly supervised label can be different even for the same sentence



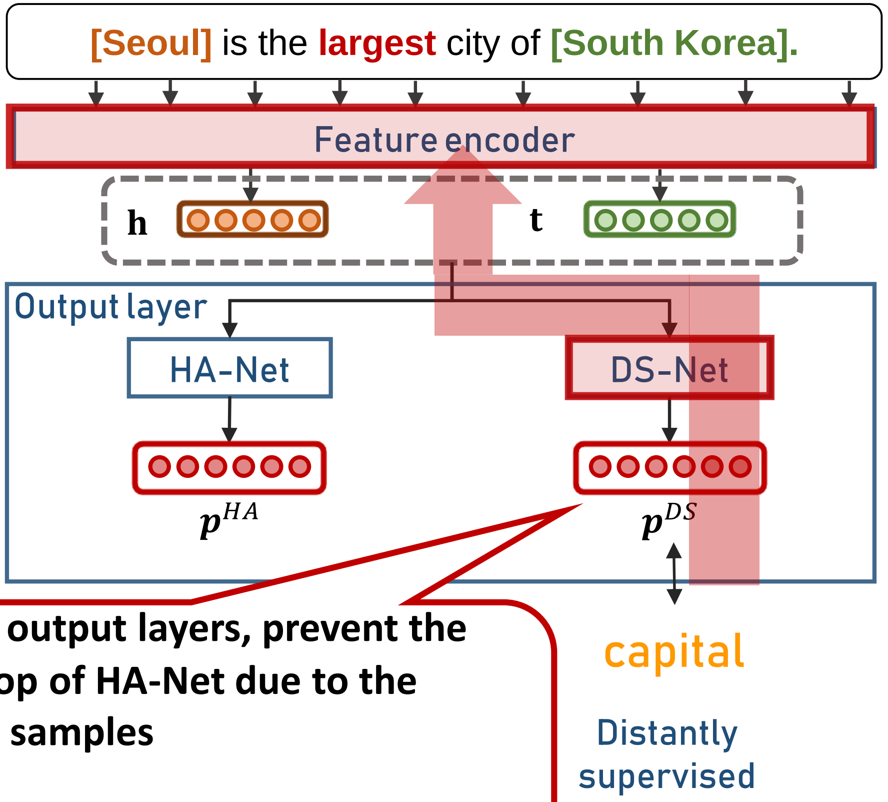
- A feature encoder
- **A single prediction network**

- Loss function
  - Cross-entropy

Distant supervision can degrade the performance of relation extraction

Dual Supervision Framework [COLING 2020]

- Dual Supervision Framework



Using separate output layers, prevent the performance drop of HA-Net due to the wrongly labeled samples

However, HA-Net cannot effectively learn from distantly supervised labels

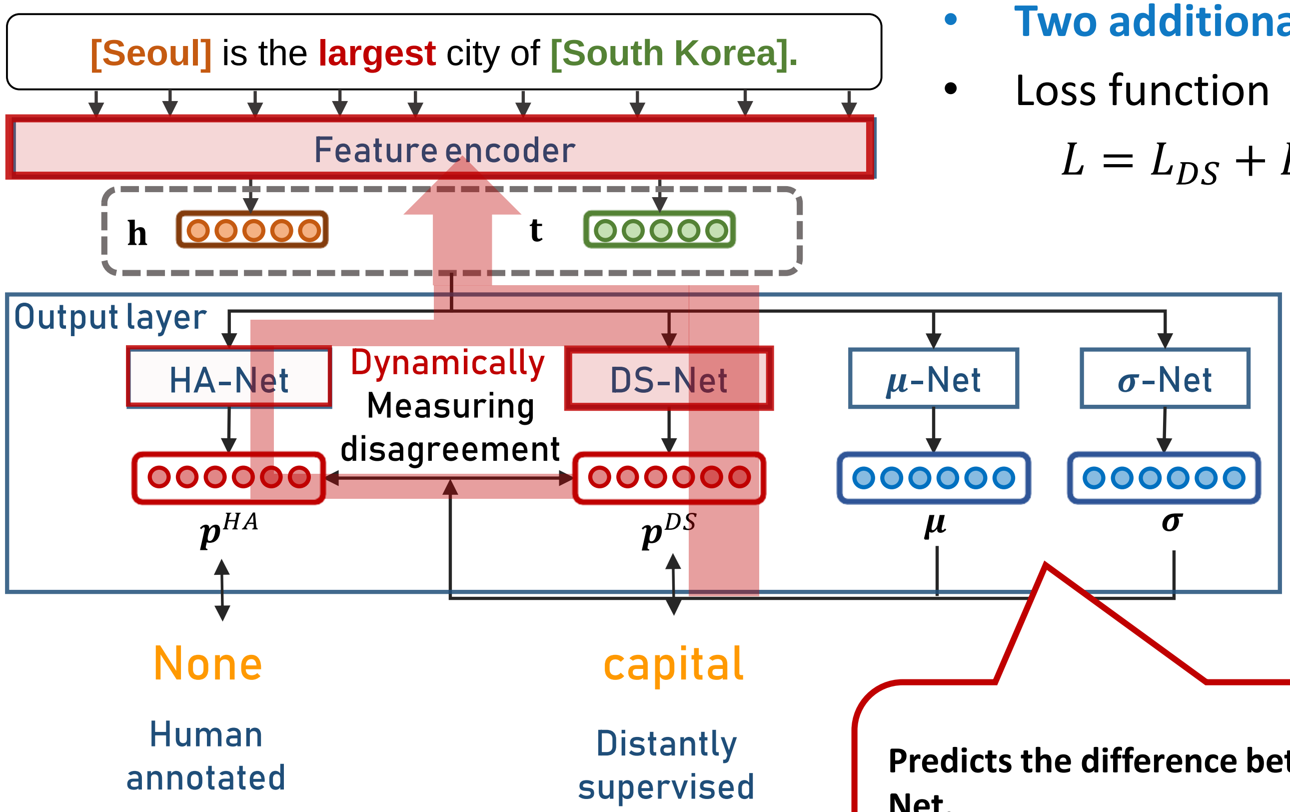
- A feature encoder
- **Separate two prediction networks**
  - HA-Net: Trained by human annotated labels
  - DS-Net: Trained by distantly supervised labels

- Loss function

$$L = L_{DS} + L_{HA}$$

L_{DS}: loss for distant supervision
L_{HA}: loss for human annotation

Dual Supervision Framework [COLING 2020]



- **Separate two prediction networks**
- **Two additional networks**
- Loss function

$$L = L_{DS} + L_{HA} + \lambda \cdot \underline{L_{DS-HA}}$$

Disagreement penalty

The difference between output probabilities from HA-Net and DS-Net should be similar to the distribution generated by μ and σ

Predicts the difference between HA-Net and DS-Net.
Considers the accuracy of DS-Net under the context and improves the performance of HA-Net.

Dual Supervision Framework [COLING 2020]

- Measuring the disagreement
  - Assume that the inflation X_r for a relation r follows $LogNormal(\mu_r, \sigma_r)$

$$P(X_r = x) = \frac{1}{x\sigma_r\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\log x - \mu_r}{\sigma_r}\right)^2\right)$$

  - The **disagreement penalty** is the negative log-likelihood of X

$$L_{DS-HA} = -\log P(p_r^{DS}/p_r^{HA})$$

$$= \frac{1}{2}\left(\frac{\log \boxed{p_r^{DS}} - \log \boxed{p_r^{HA}} - \boxed{\mu_r}}{\boxed{\sigma_r}}\right)^2 + \log \boxed{p_r^{DS}} - \log \boxed{p_r^{HA}} + \log \boxed{\sigma_r}$$

Module	HA-Net	DS-Net	μ-Net	σ-Net
Output	p_r^{HA}	p_r^{DS}	μ_r	σ_r

Dual Supervision Framework [COLING 2020]

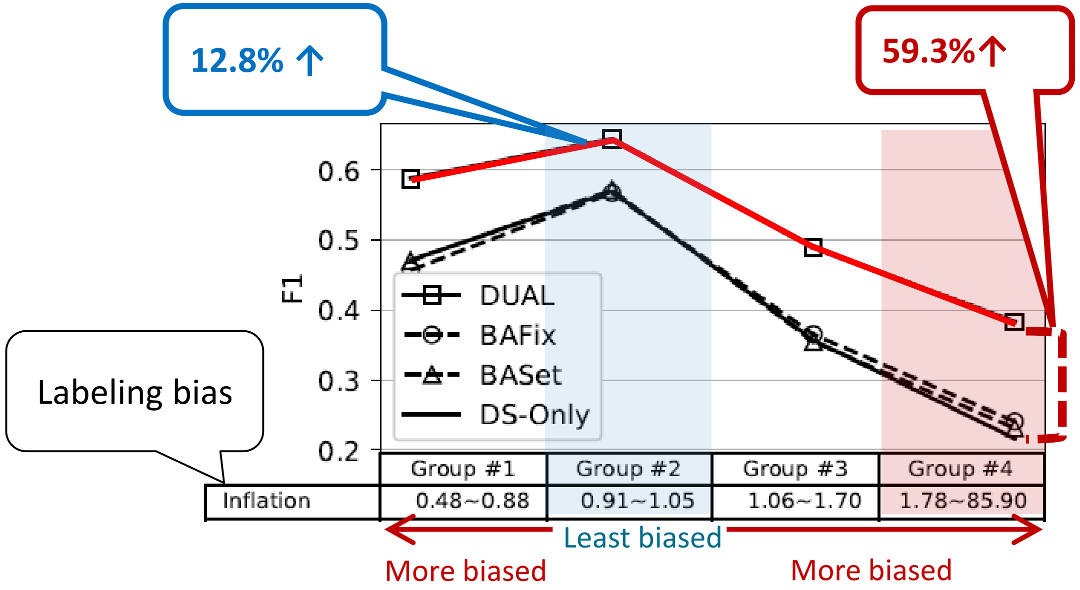
- Datasets
  - **KBP, NYT**: sentence-level relation extraction datasets
  - **DocRED**: a document-level relation extraction dataset

Data	Number of instances Train-HA	Train-DS	Dev	Test	# of rel. types
KBP	378	132,369	14,103	1,488	7
NYT	756	323,126	34,871	3,021	25
DocRED	38,269	1,508,320	12,332	12,842	96

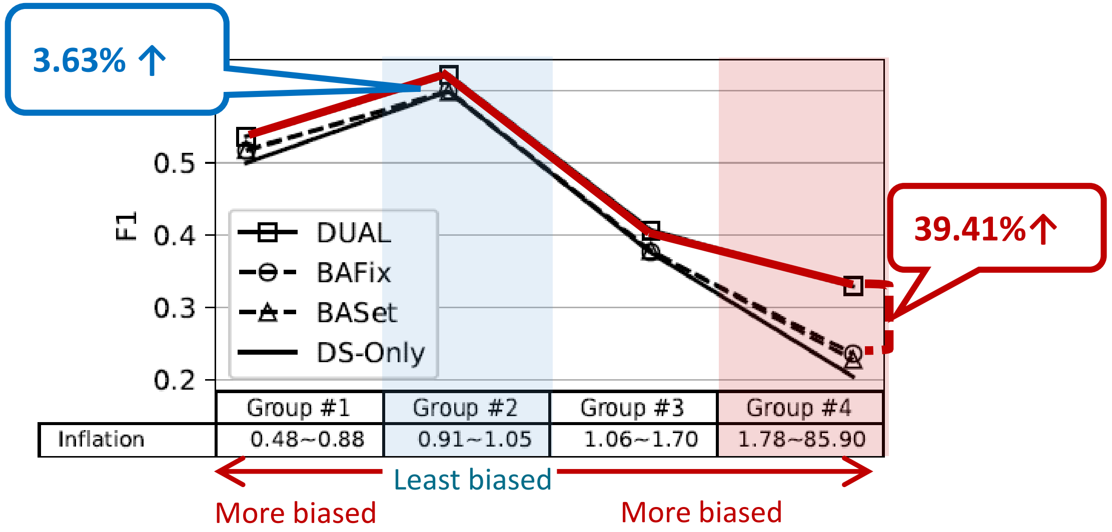
Dual Supervision Framework [COLING 2020]

- Experimental result

When DS data contains more errors, the performance gain becomes higher



BERT



BiLSTM

Relation extraction accuracy improvement

Questions