Minimally Supervised Learning from Text

Giannis Karamanolakis
Department of Computer Science, Columbia University
gkaraman@cs.columbia.edu

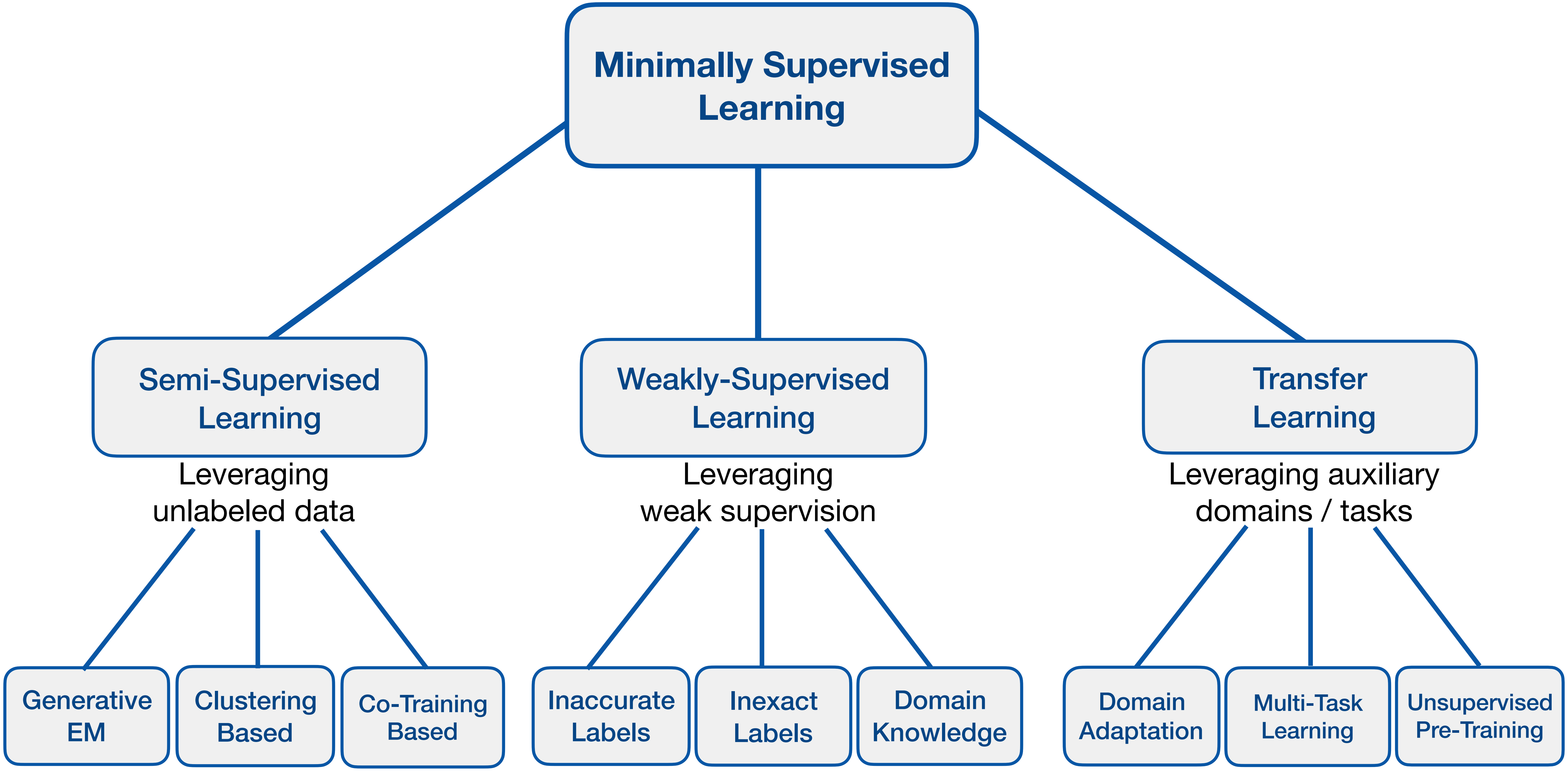
Candidacy Exam

April 6th, 2020

Committee: Michael Collins, Luis Gravano, Daniel Hsu

COLUMBIA UNIVERSITY

Taxonomy


Minimally Supervised Learning
Semi-Supervised Learning
Leveraging unlabeled data
Weakly-Supervised Learning
Leveraging weak supervision
Transfer Learning
Leveraging auxiliary domains / tasks
Generative EM
Clustering Based
Co-Training Based
Inaccurate Labels
Inexact Labels
Domain Knowledge
Domain Adaptation
Multi-Task Learning
Unsupervised Pre-Training

Problem of Focus - Text Classification

- **Goal:** classify input text (e.g., document, sentence, clause, …)
  to pre-defined target classes (e.g., positive/negative sentiment)

<u>**input text**</u> x		<u>**target class**</u> y
"Totally dissatisfied with the service"	→	*Negative Sentiment*

- **Applications:**
  - Sentiment/emotion classification (e.g., Yelp, IMDB, Amazon, Twitter)
  - Categorization of news/financial documents (e.g., Reuters, Wall Street Journal)
  - Spam/fraud detection (e.g., Yahoo, Outlook)
  - User intent detection (e.g., Gmail, Siri, Alexa)
  - Emergency detection (e.g., earthquake, outbreaks)
  - …

Text Classification - Approaches Over Time

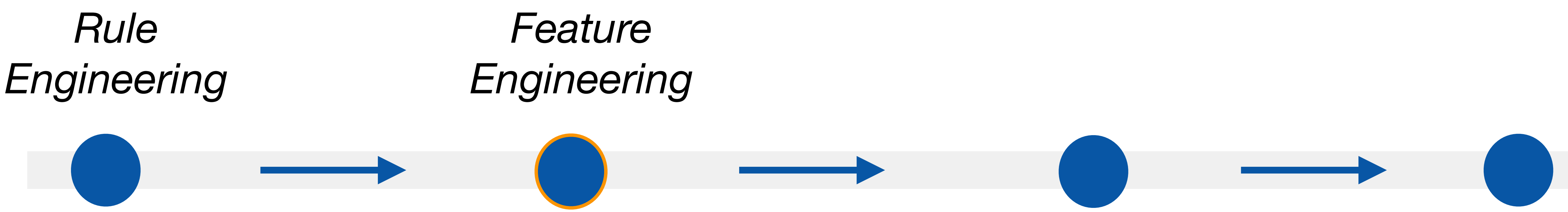
Rule Engineering

Use rules, hard-coded by humans

(—) limited generalization

$$x \xrightarrow{\textit{rules}} y$$

Text Classification - Approaches Over Time



Automatically learn "rules" from labeled data

… via supervised learning

Focus: Find good "features" for x

(—) time-consuming

tf-idf, POS tags, parse-trees, …

$$x \xrightarrow{f} y$$

Text Classification - Approaches Over Time

Rule Engineering

Feature Engineering

Model Architecture Engineering

Automatically learn features from data

… via supervised deep learning

$$x \xrightarrow{f} y$$

Text Classification - Approaches Over Time

Rule Engineering → *Feature Engineering* → *Model Architecture Engineering* →

Automatically learn features from data

… via supervised deep learning

Focus: Find good model architectures f

CNNs, RNNs, Transformers, …

$$x \xrightarrow{f} y$$

Text Classification - Approaches Over Time

Rule Engineering → *Feature Engineering* → *Model Architecture Engineering* →

Automatically learn features from data

… via supervised deep learning

Focus: Find good model architectures f

(+) high predictive accuracy
(—) “data-hungry”

CNNs, RNNs, Transformers, …

$$x \xrightarrow{f} y$$

Text Classification - Approaches Over Time

Rule Engineering → *Feature Engineering* → *Model Architecture Engineering* →

Data Annotation Bottleneck in Supervised Learning

- Requires many **ground-truth** annotations

$$D_L = \{(x_i, \cancel{y_i})\}_{i=1}^{N}$$

- Manual annotation is **expensive** and **time-consuming**

(—) "data-hungry"

$$x \xrightarrow{f} y$$

Supervision Engineering
Learning With Limited Labeled Data

Rule Engineering → *Feature Engineering* → *Model Architecture Engineering* → *Supervision Engineering*

Leverage cheaper types of supervision
… for training machine learning models

$$D_L = \{(x_i, \cancel{y_i})\}_{i=1}^{N}$$

???

$$x \xrightarrow{f} y$$

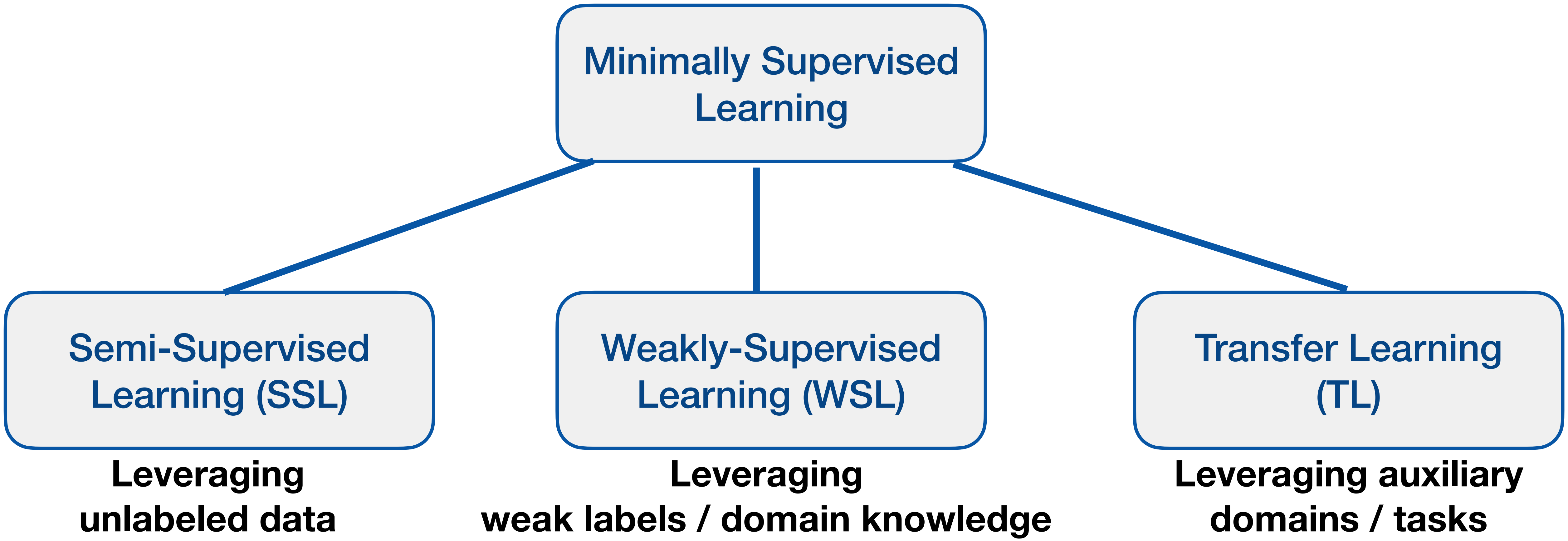
This presentation

An overview of approaches for supervision engineering

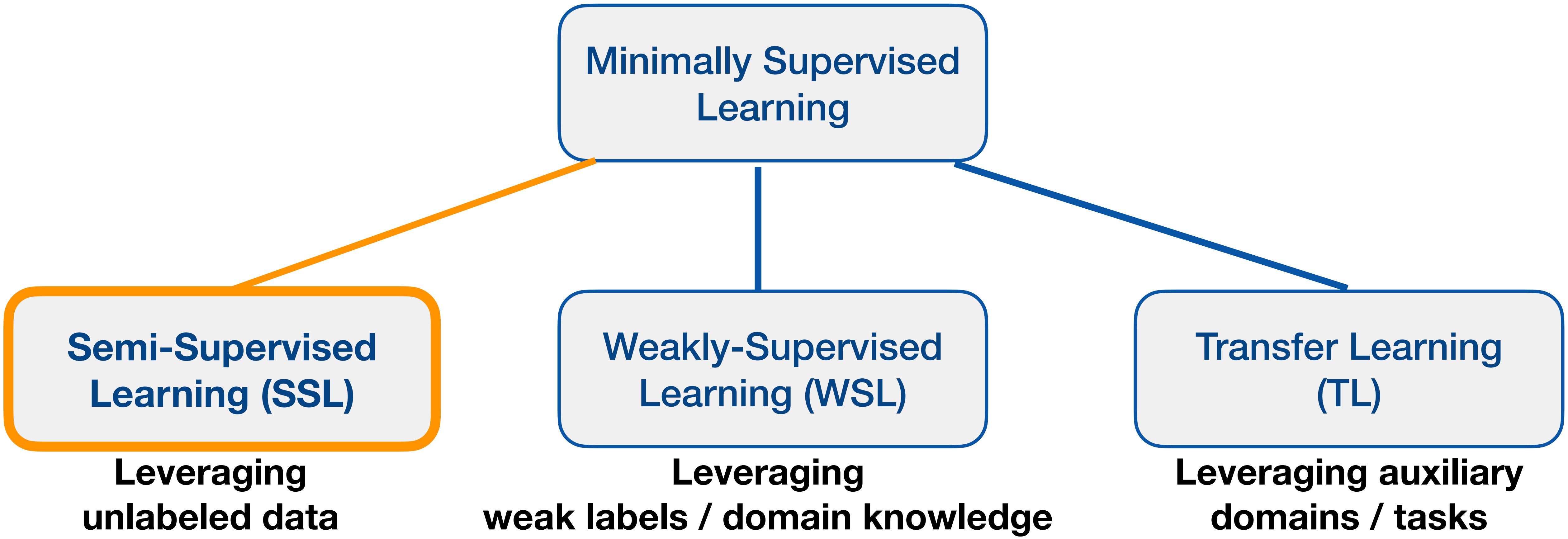
???

$$x \xrightarrow{f} y$$

Taxonomy


Minimally Supervised Learning
Semi-Supervised Learning (SSL)
Weakly-Supervised Learning (WSL)
Transfer Learning (TL)
Leveraging unlabeled data
Leveraging weak labels / domain knowledge
Leveraging auxiliary domains / tasks

Taxonomy



[Nigam et al., 1999]
[Joachims, 1999]
[Blum & Mitchell, 1998]
[Nigam & Ghani, 2000]
[Zhu et al., 2000]
[Seeger, 2006]
[Clark et al., 2018]
[Ruder & Plank, 2018]

SSL - Leveraging Unlabeled Data

- **Semi-Supervised Learning (SSL):**
  - Small number of labeled data:

$$D_L = \{(x_i, y_i)\}_{i=1}^{N}$$

expensive

  - … and large number of unlabeled data:

$$D_U = \{x_i\}_{i=N+1}^{M}$$

cheap

SSL - Leveraging Unlabeled Data

- **Semi-Supervised Learning (SSL):**
  - Small number of labeled data:

$$D_L = \{(x_i, y_i)\}_{i=1}^{N}$$

expensive

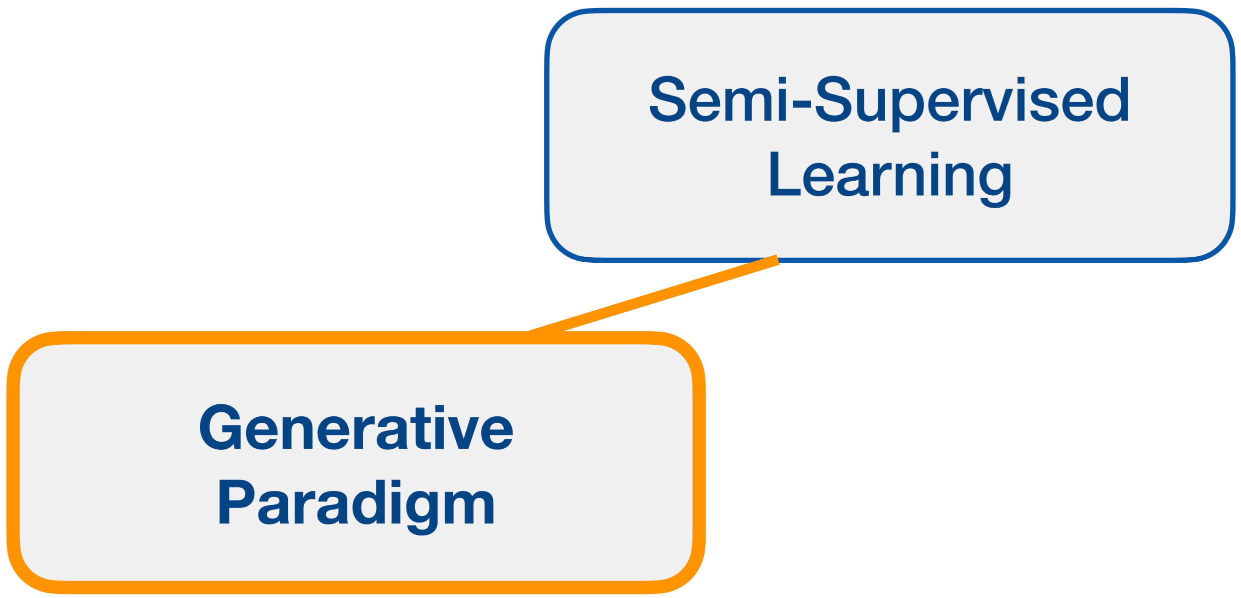
  - … and large number of unlabeled data:

$$D_U = \{x_i\}_{i=N+1}^{M}$$

cheap

- **SSL goal:**
  - Learn $f : x \rightarrow y$
  - … by leveraging $D_L + D_U$
  - … **more effectively** than using just D_L

SSL Taxonomy



Leveraging Unlabeled Data - Generative Modeling Approach

- Use D_U to determine a better **generative model** $P(X, Y)$ [Nigam et al., 1999]
  - Unobserved labels: **missing values**
  - Learning e.g., via Expectation-Maximization (EM)

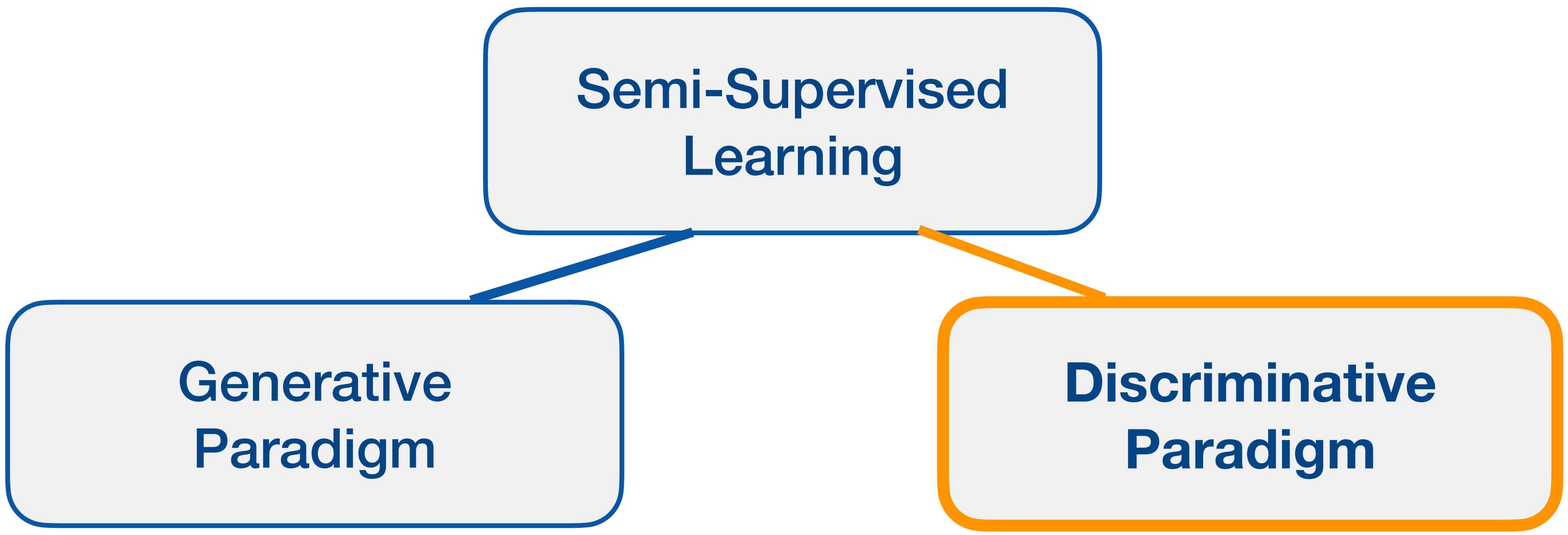
Leveraging Unlabeled Data - Generative Modeling Approach

- Use D_U to determine a better **generative model** $P(X, Y)$ [Nigam et al., 1999]
  - Unobserved labels: **missing values**
  - Learning e.g., via Expectation-Maximization (EM)

(-) misspecification issues:

if modeling assumptions != natural data distribution performance may suffer

SSL Taxonomy



Leveraging Unlabeled Data - Discriminative Modeling Approaches

- Use D_U to determine a better **discriminative model** $P(Y|X)$

Leveraging Unlabeled Data - Discriminative Modeling Approaches

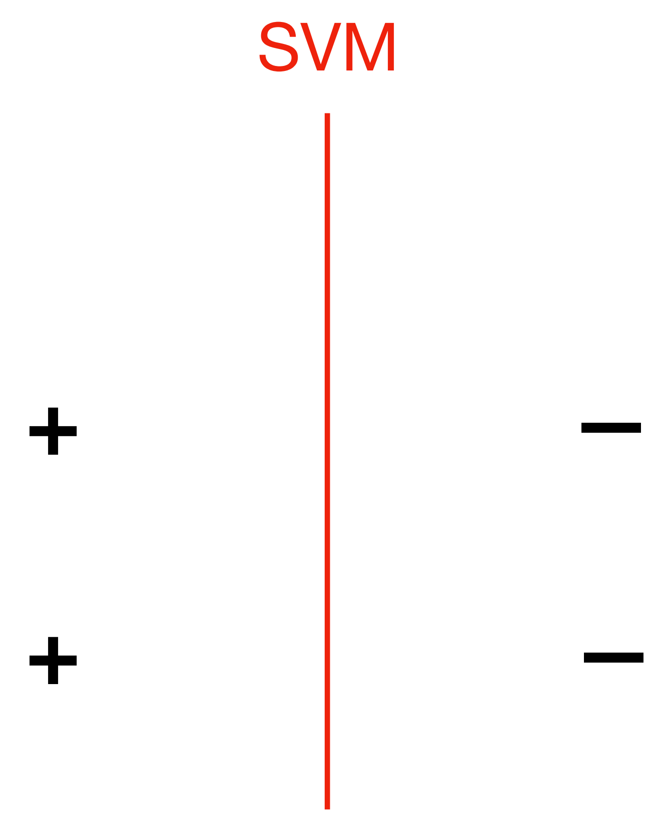
- Use D_U to determine a better **discriminative model** $P(Y|X)$

    Need assumptions: “When are unlabeled examples informative?”

Leveraging Unlabeled Data - Discriminative Modeling Approaches

- Use D_U to determine a better **discriminative model** $P(Y|X)$

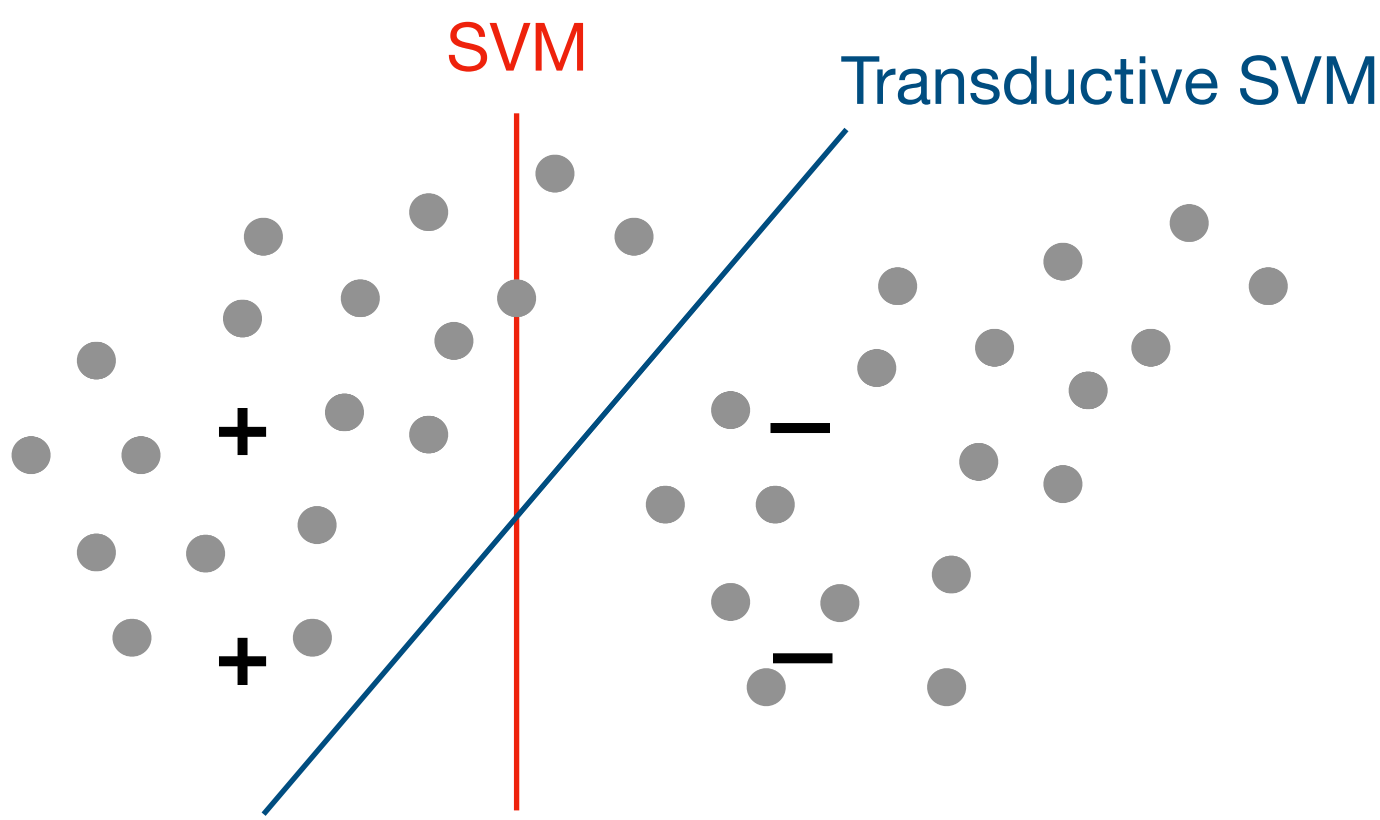
  Need assumptions: “When are unlabeled examples informative?”



Leveraging Unlabeled Data - Discriminative Modeling Approaches

- Use D_U to determine a better **discriminative model** $P(Y|X)$

  Need assumptions: “When are unlabeled examples informative?”

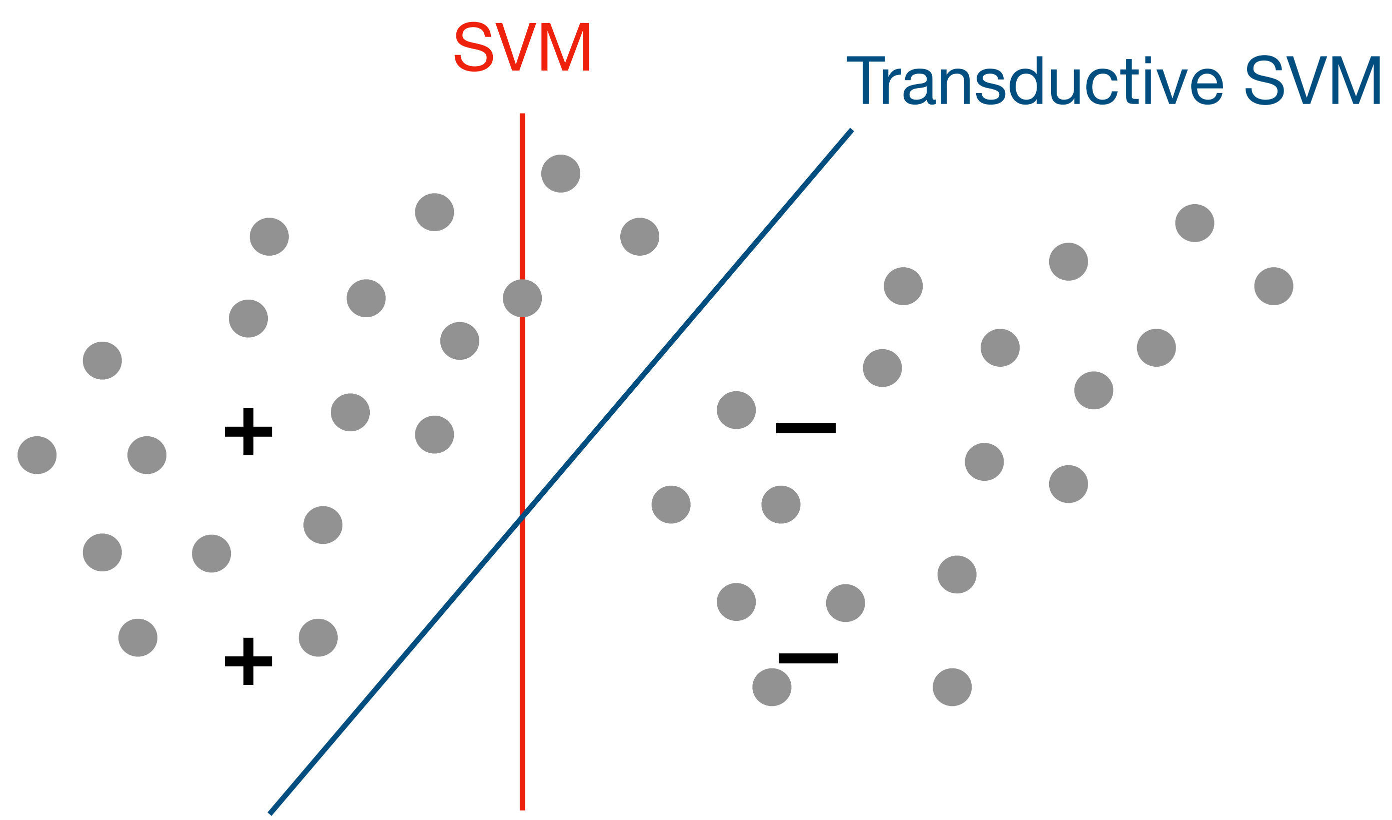


[Joachims, 1999]

Leveraging Unlabeled Data - Discriminative Modeling Approaches

- Use D_U to determine a better **discriminative model** $P(Y|X)$

  Need assumptions: "When are unlabeled examples informative?"



[Joachims, 1999]

"clustering assumption"

- **graph-based:** label propagation from labeled to unlabeled wrt. similarity

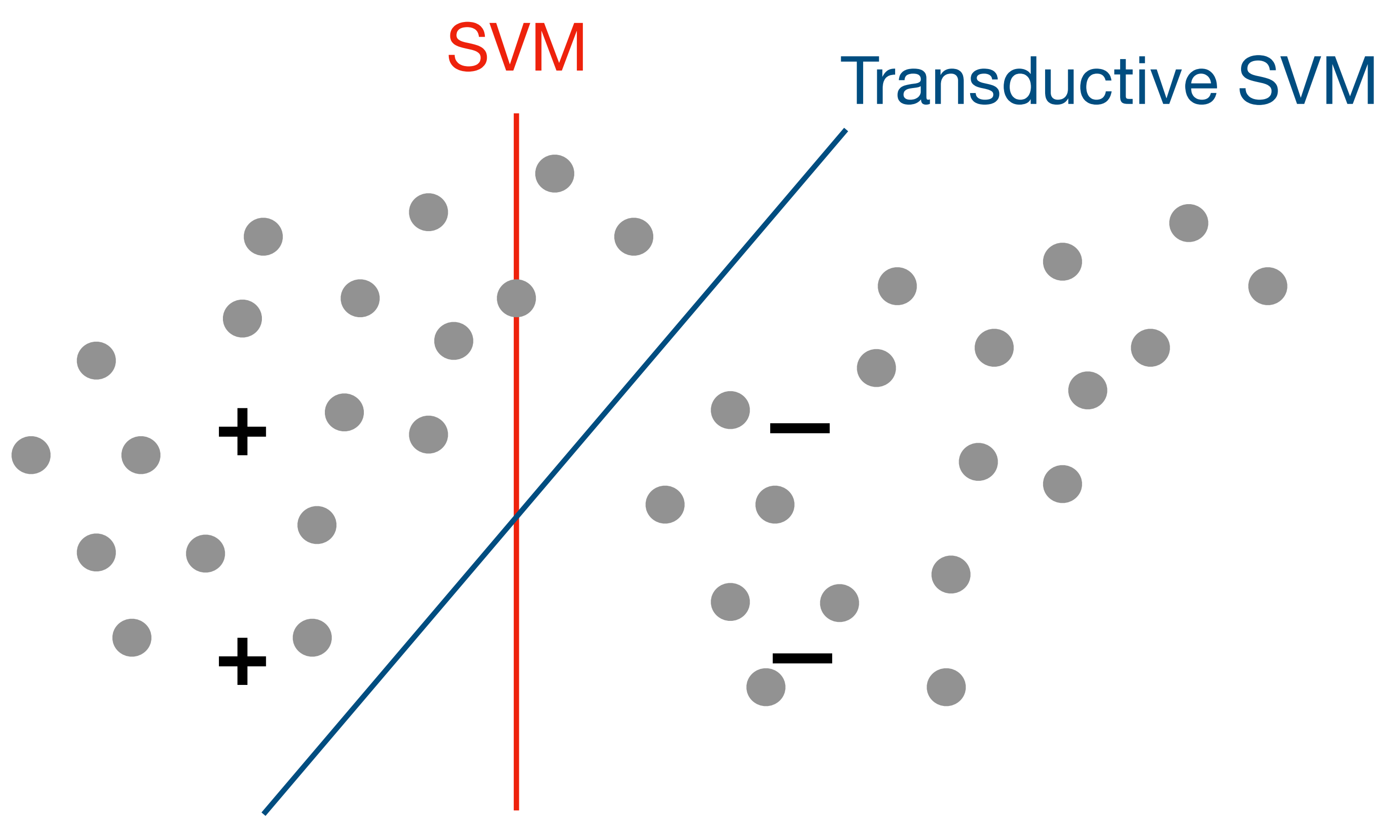
[Zhu et al., 2000]

Leveraging Unlabeled Data - Discriminative Modeling Approaches

- Use D_U to determine a better **discriminative model** $P(Y|X)$

  Need assumptions: “When are unlabeled examples informative?”

[Joachims, 1999]



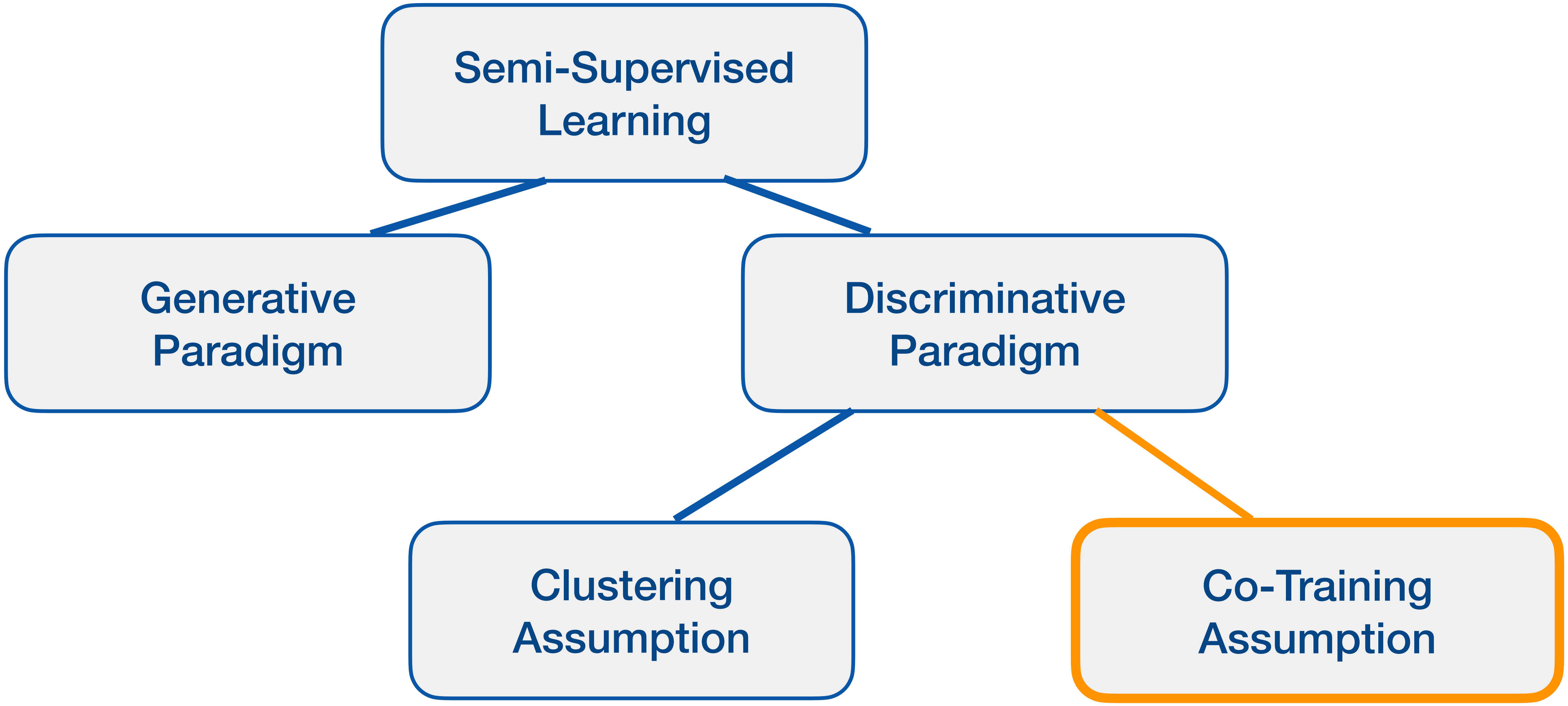
“clustering assumption”

- **graph-based:** label propagation from labeled to unlabeled wrt. similarity

[Zhu et al., 2000]

(-) scalability issues

SSL Taxonomy



Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple “views"**

text

hyperlinks pointing to page

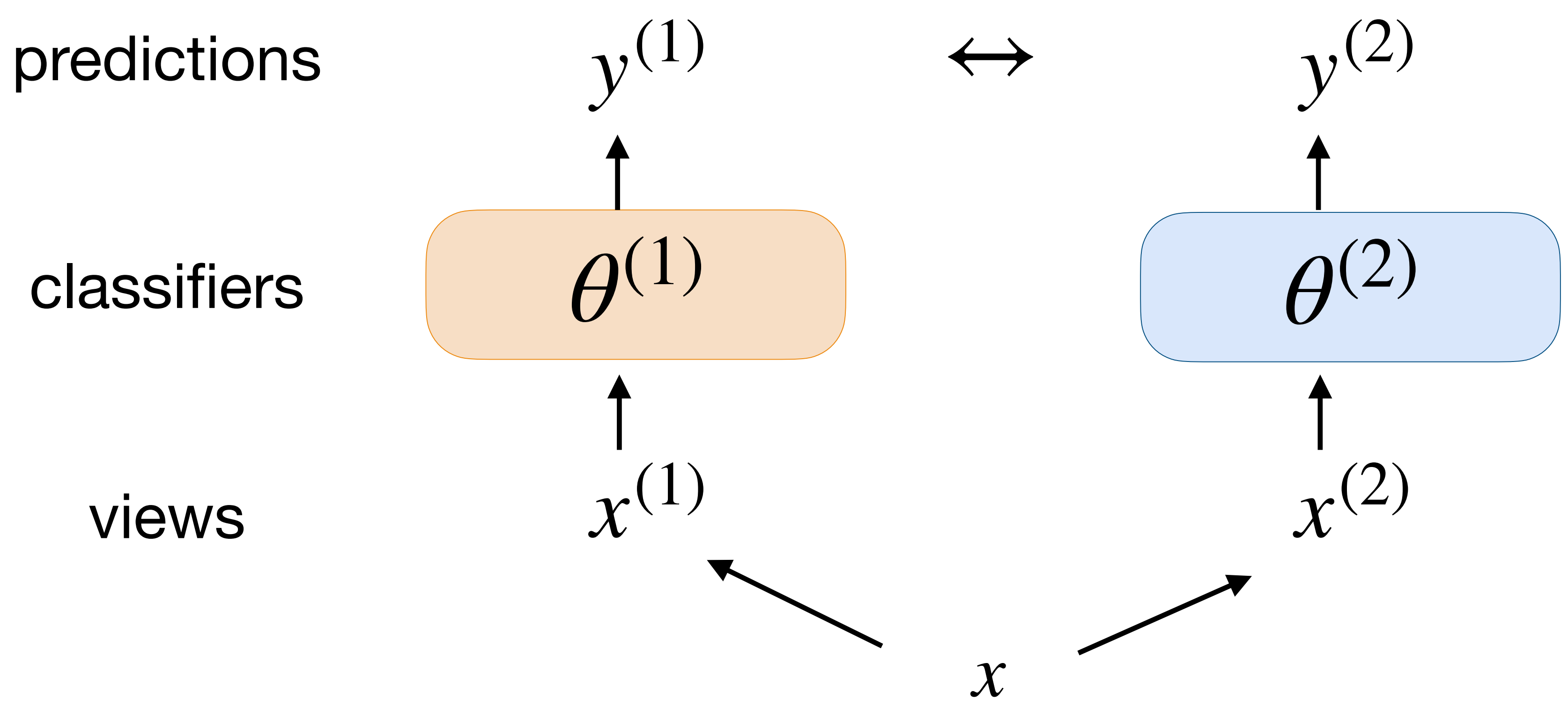
views $x^{(1)}$ $x^{(2)}$

$$x^{(1)} \leftarrow x \rightarrow x^{(2)}$$

academic webpages

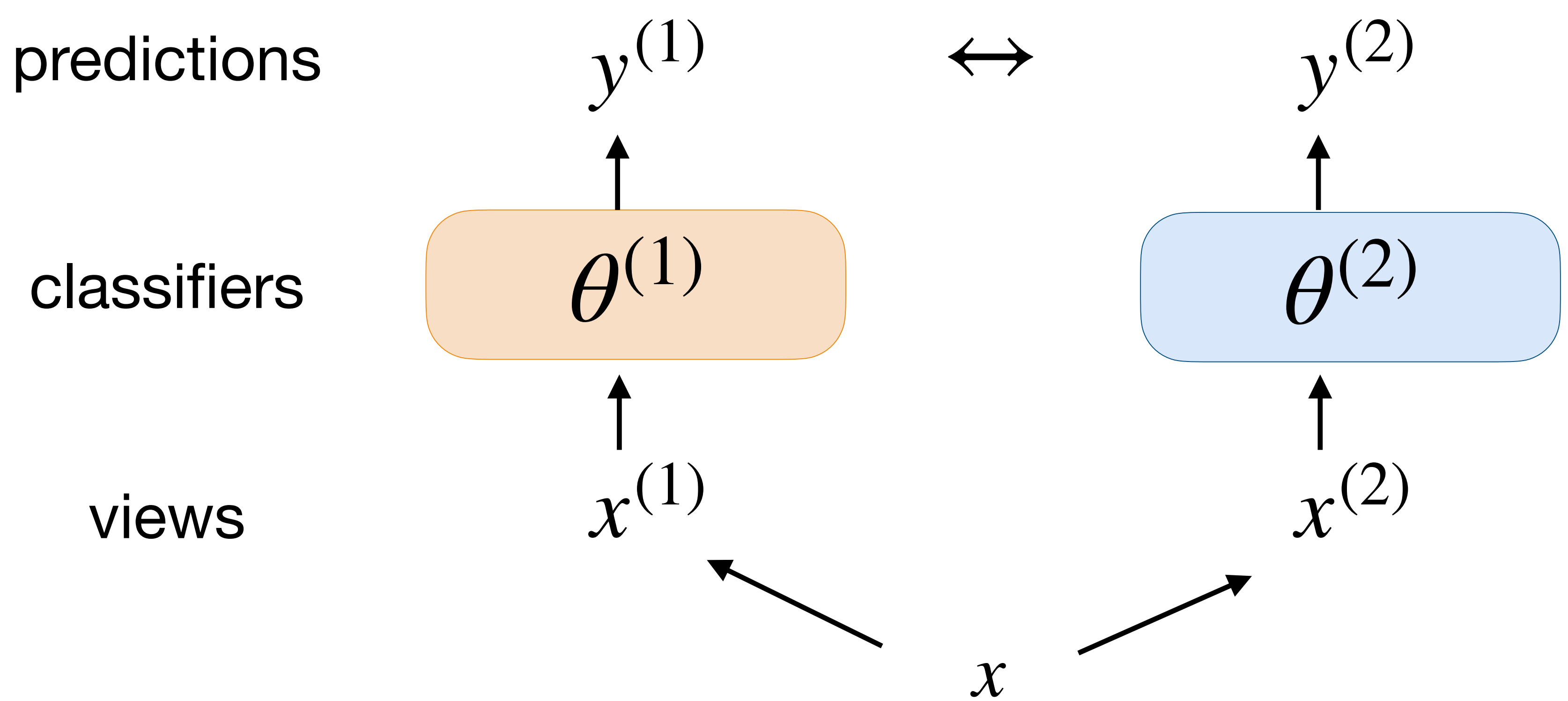
Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple "views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}$, $\theta^{(2)}$ each considering a **different** view $x^{(1)}$, $x^{(2)}$



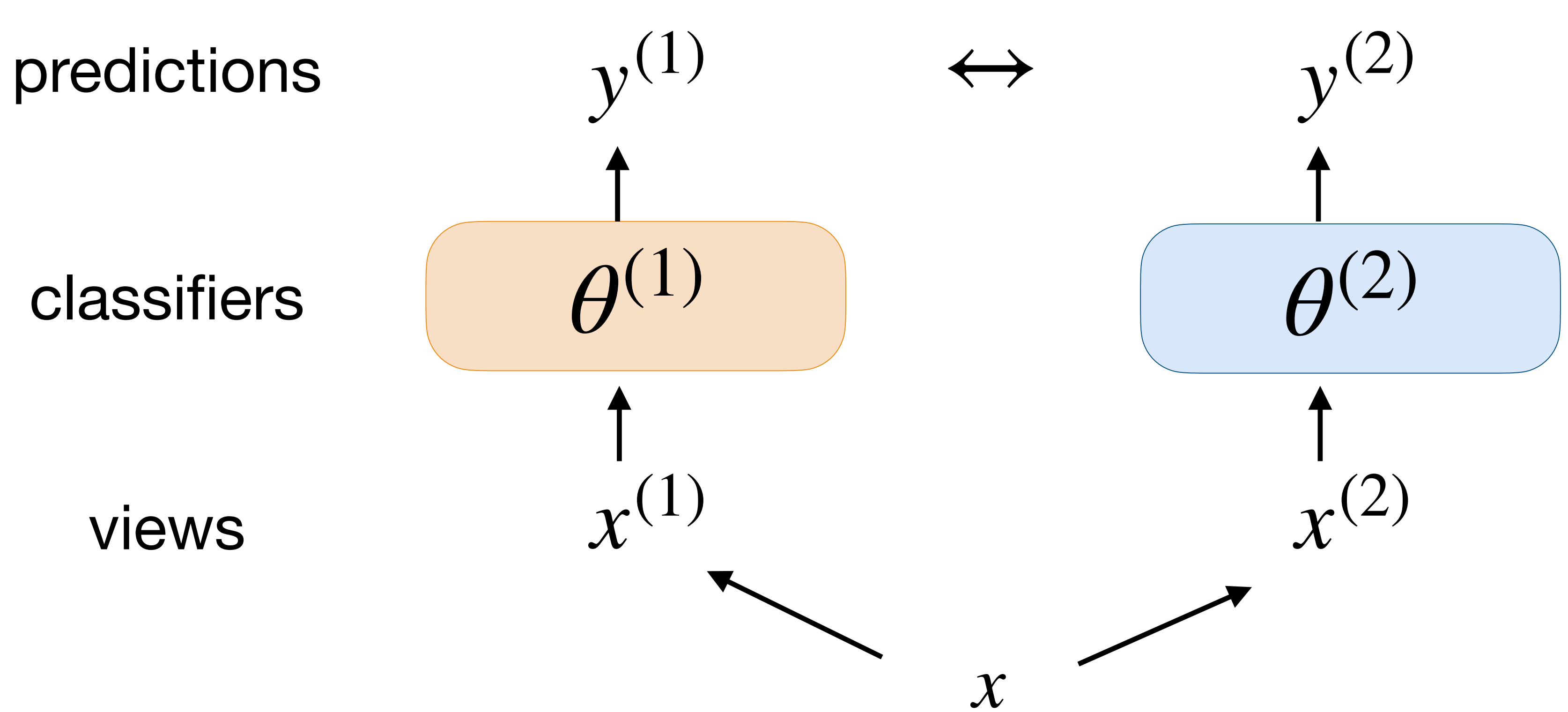
Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple “views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}$, $\theta^{(2)}$ each considering a **different** view $x^{(1)}$, $x^{(2)}$
  - Goal: maximize **agreement** between $y^{(1)}$, $y^{(2)}$ on **unlabeled** data D_U



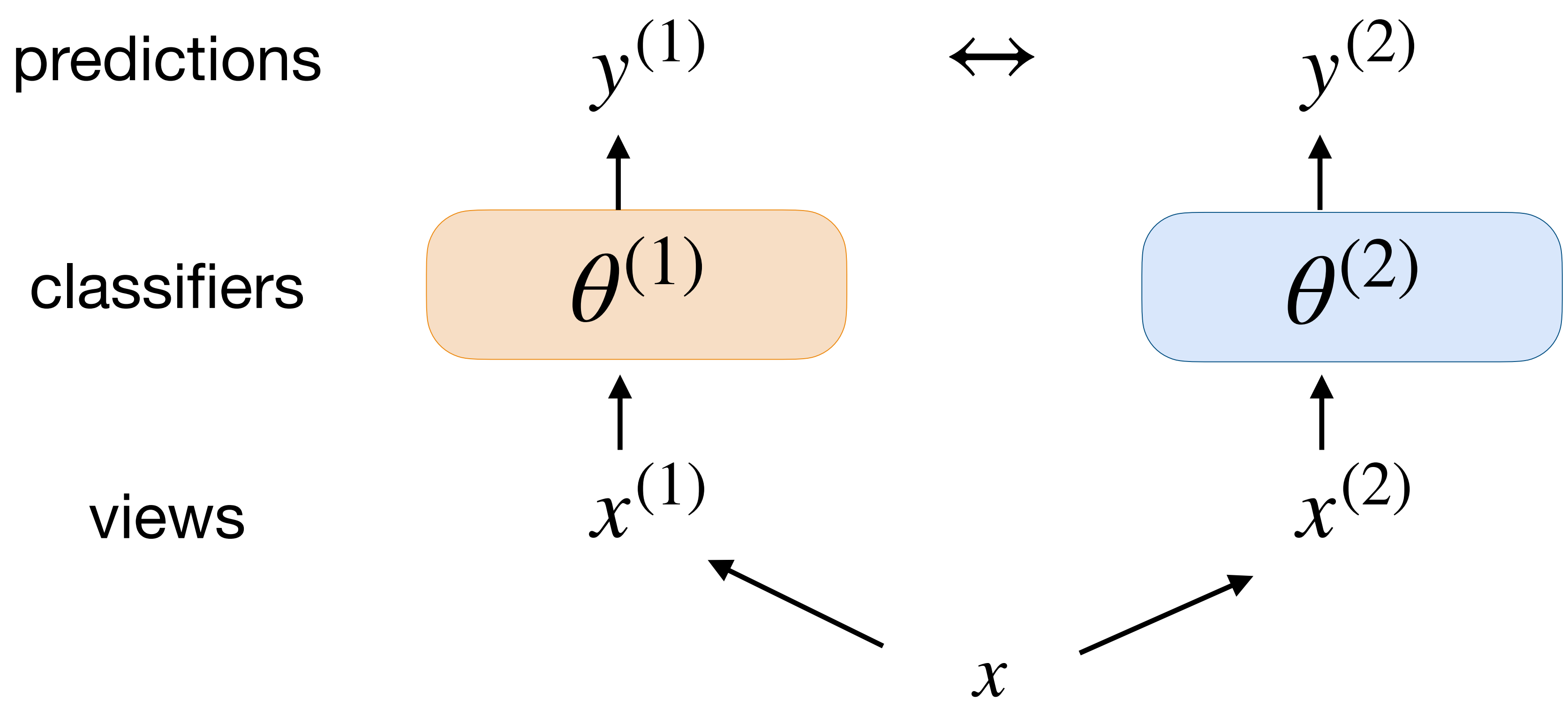
Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple “views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}, \theta^{(2)}$ each considering a **different** view $x^{(1)}, x^{(2)}$
  - Goal: maximize **agreement** between $y^{(1)}, y^{(2)}$ on **unlabeled** data D_U
  - How: confident $y^{(1)}$ on **unlabeled** D_U used as **extra training data** for $\theta^{(2)}$



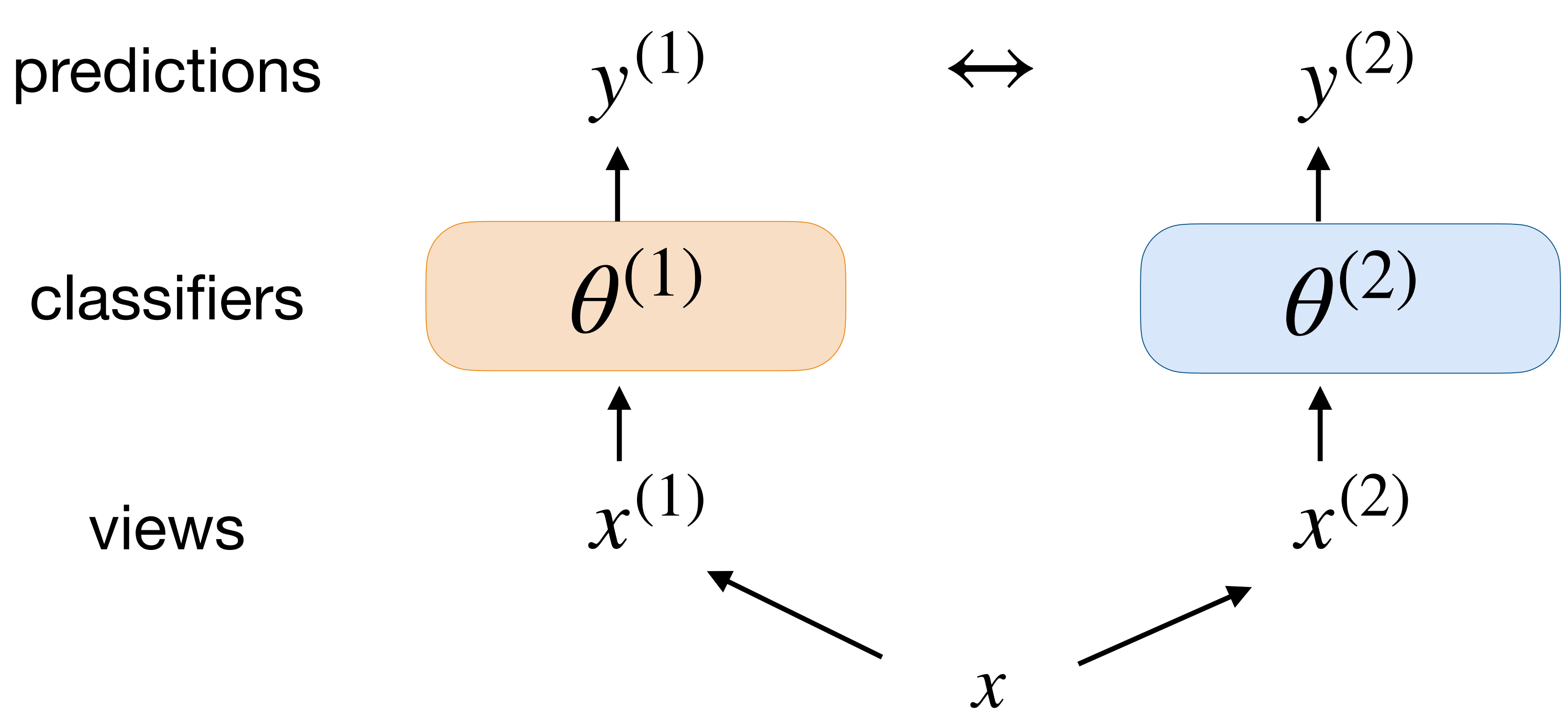
Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple “views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}, \theta^{(2)}$ each considering a **different** view $x^{(1)}, x^{(2)}$
  - Goal: maximize **agreement** between $y^{(1)}, y^{(2)}$ on **unlabeled** data D_U
  - How: confident $y^{(1)}$ on **unlabeled** D_U used as **extra training data** for $\theta^{(2)}$
  - Maximum benefit when sufficiently **diverse** views: **“conditional independence”**



Co-Training for Multi-View Learning

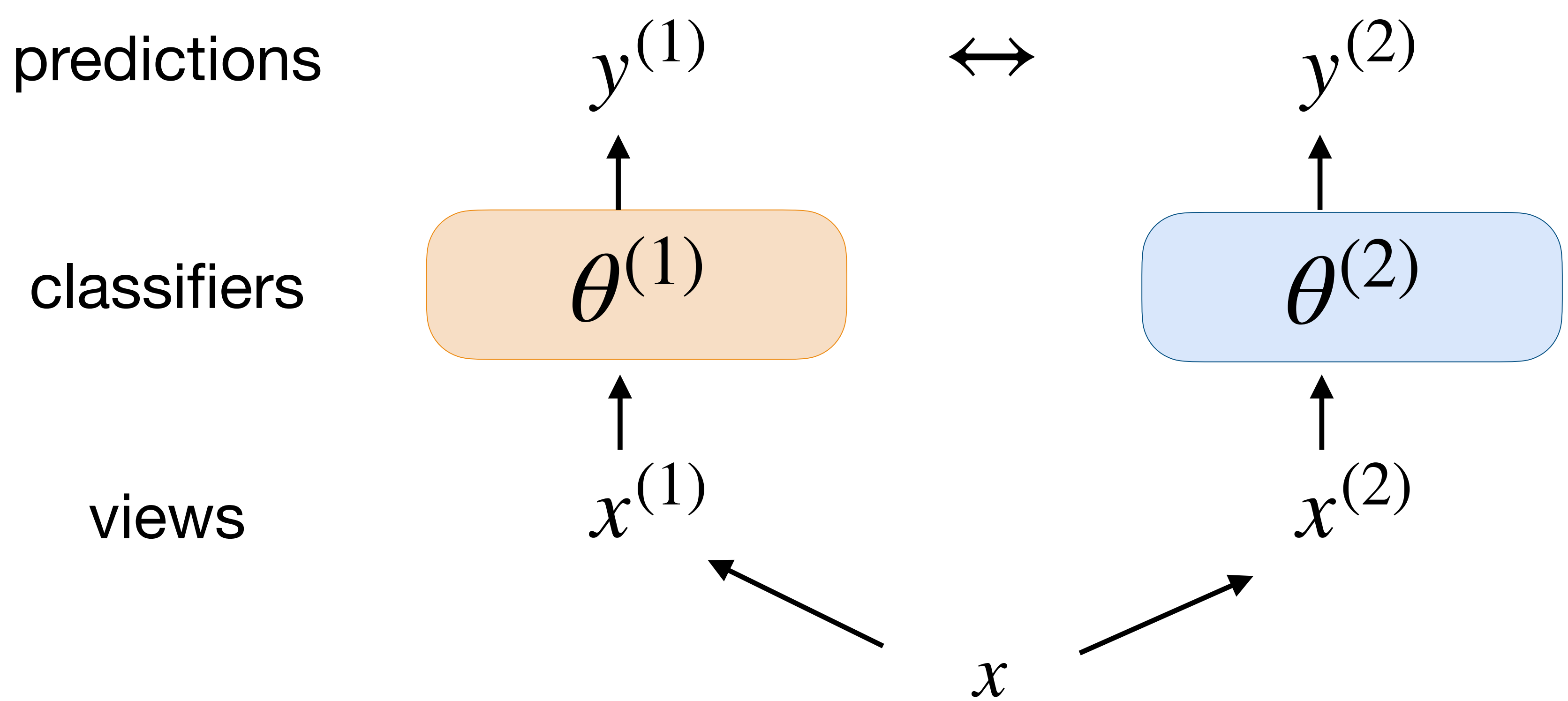
- **Observation:** sometimes examples could be described by **multiple "views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}$, $\theta^{(2)}$ each considering a **different** view $x^{(1)}$, $x^{(2)}$
  - Goal: maximize **agreement** between $y^{(1)}$, $y^{(2)}$ on **unlabeled** data D_U
  - How: confident $y^{(1)}$ on **unlabeled** D_U used as **extra training data** for $\theta^{(2)}$
  - Maximum benefit when sufficiently **diverse** views: **"conditional independence"**



(-) strong assumption: conditional independence unlikely to be satisfied in practice.

Co-Training for Multi-View Learning

- **Observation:** sometimes examples could be described by **multiple "views"**
- **Co-Training:** agreement-based [Blum & Mitchell, 1998]
  - Setting: **two** classifiers $\theta^{(1)}$, $\theta^{(2)}$ each considering a **different** view $x^{(1)}$, $x^{(2)}$
  - Goal: maximize **agreement** between $y^{(1)}$, $y^{(2)}$ on **unlabeled** data D_U
  - How: confident $y^{(1)}$ on **unlabeled** D_U used as **extra training data** for $\theta^{(2)}$
  - Maximum benefit when sufficiently **diverse** views: **"conditional independence"**



(-) strong assumption: conditional independence unlikely to be satisfied in practice.

But, does it really need to be satisfied?

Extending Co-Training to More Practical Settings

- Further work: **good performance** even if assumptions are violated!

[Nigam & Ghani, 2000]

Extending Co-Training to More Practical Settings

- Further work: **good performance** even if assumptions are violated!
  [Nigam & Ghani, 2000]
- Co-training **could** be applied even in **single-view** settings:
  - Divide features in **diverse** views:
    - “spelling” & “contextual” features of named entities [Collins & Singer, 1999]
    - Internal (forward and backward) features of a BiLSTM [Clark et al., 2018]

Extending Co-Training to More Practical Settings

- Further work: **good performance** even if assumptions are violated!

  [Nigam & Ghani, 2000]

- Co-training **could** be applied even in **single-view** settings:
  - Divide features in **diverse** views:
    - "spelling" & "contextual" features of named entities [Collins & Singer, 1999]
    - Internal (forward and backward) features of a BiLSTM [Clark et al., 2018]
  - Use **diverse** architectures:
    - 3 diverse classifiers: add training data to θ_1 if θ_2, θ_3 agree [Zhou et al., 2005]

Extending Co-Training to More Practical Settings

- Further work: **good performance** even if assumptions are violated!

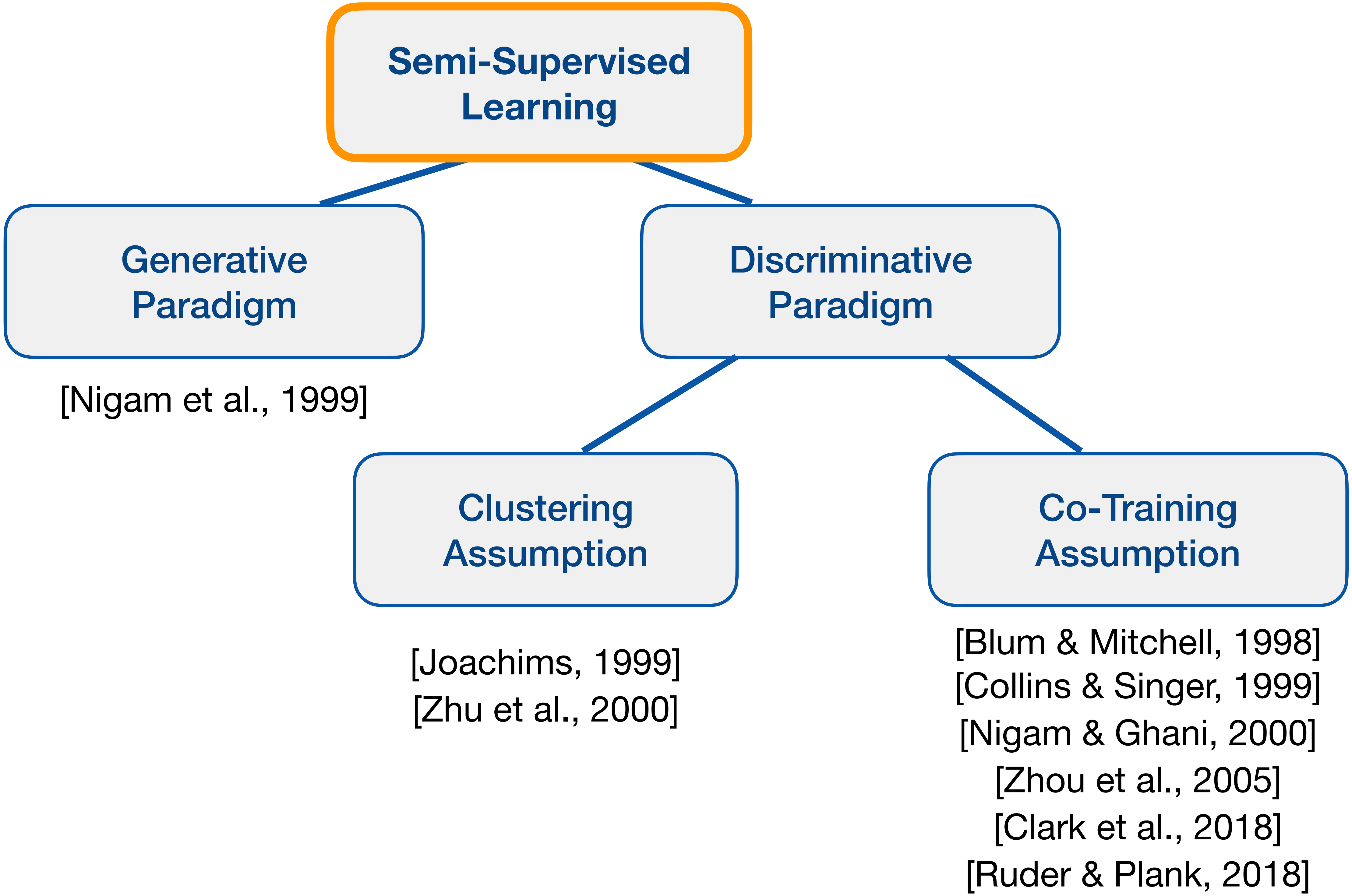
  [Nigam & Ghani, 2000]

- Co-training **could** be applied even in **single-view** settings:
  - Divide features in **diverse** views:
    - “spelling” & “contextual” features of named entities [Collins & Singer, 1999]
    - Internal (forward and backward) features of a BiLSTM [Clark et al., 2018]
  - Use **diverse** architectures:
    - 3 diverse classifiers: add training data to θ_1 if θ_2, θ_3 agree [Zhou et al., 2005]
      - Still a strong baseline in 2018! [Ruder & Plank, 2018]

Extending Co-Training to More Practical Settings

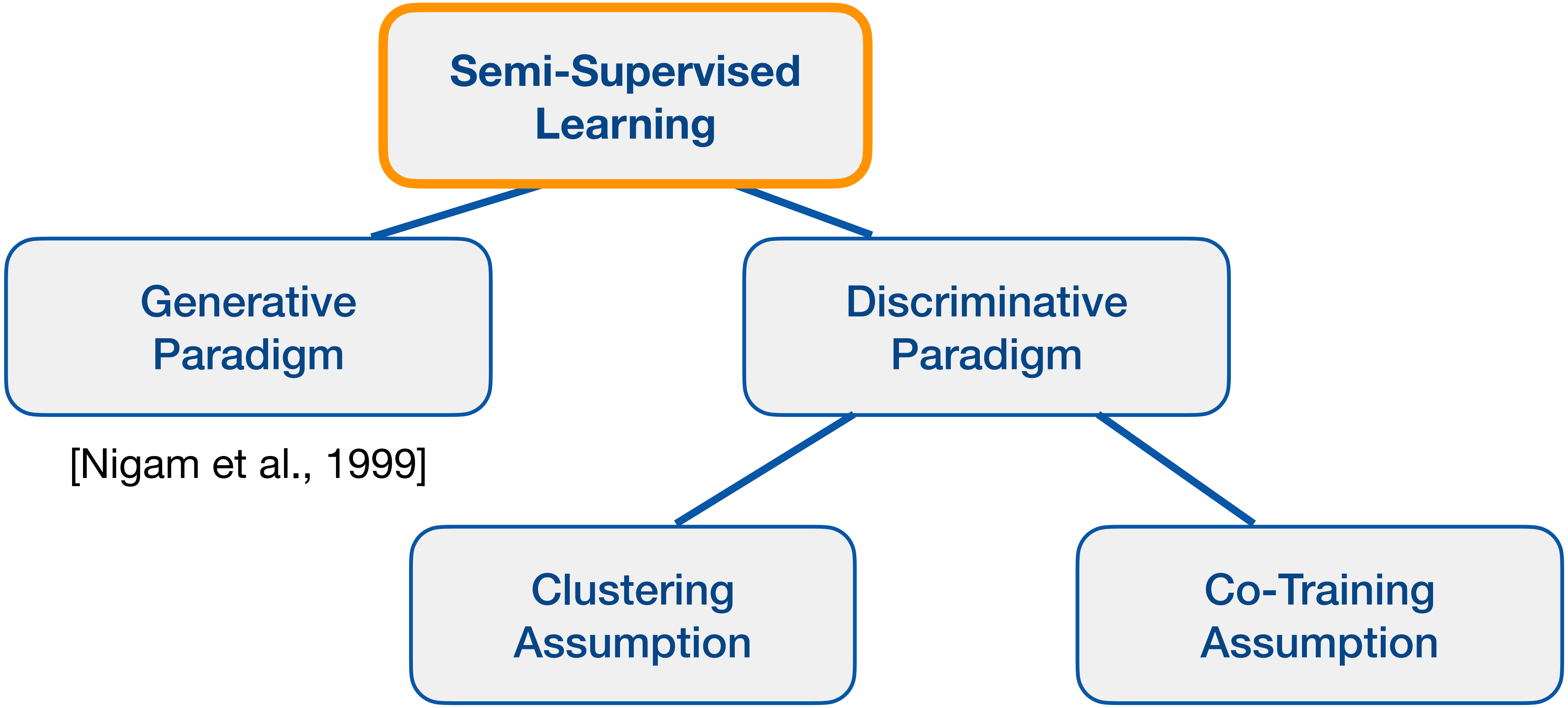
- Further work: **good performance** even if assumptions are violated!
  [Nigam & Ghani, 2000]
- Co-training **could** be applied even in **single-view** settings:
  - Divide features in **diverse** views:
    - "spelling" & "contextual" features of named entities [Collins & Singer, 1999]
    - Internal (forward and backward) features of a BiLSTM [Clark et al., 2018]
  - Use **diverse** architectures:
    - 3 diverse classifiers: add training data to θ_1 if θ_2, θ_3 agree [Zhou et al., 2005]
      - Still a strong baseline in 2018! [Ruder & Plank, 2018]
- Common pattern:
  - Encourage **agreement** between predictions…
  - … via **maximally diverse** views / classifiers

SSL Summary



- SSL leverages a few **ground-truth labeled** + a lot of **unlabeled data**

SSL Summary



[Nigam et al., 1999]

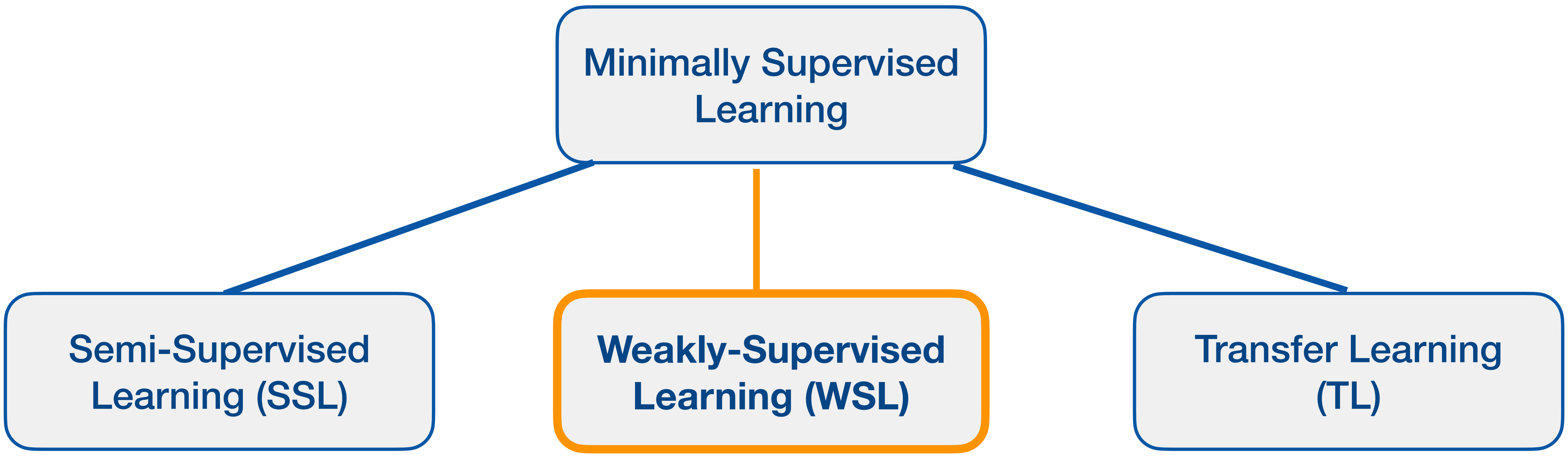
[Joachims, 1999]
[Zhu et al., 2000]

[Blum & Mitchell, 1998]
[Collins & Singer, 1999]
[Nigam & Ghani, 2000]
[Zhou et al., 2005]
[Clark et al., 2018]
[Ruder & Plank, 2018]

- SSL leverages a few **ground-truth labeled** + a lot of **unlabeled data**

(-) Limitation: not leverage information captured through other signals/metadata

Taxonomy



incomplete supervision

weak supervision

Weakly Supervised Learning (WSL)

- What is **weak** supervision?

Inaccurate labels

$$D_L = \{(x_i, y_i')\}_{i=1}^N$$

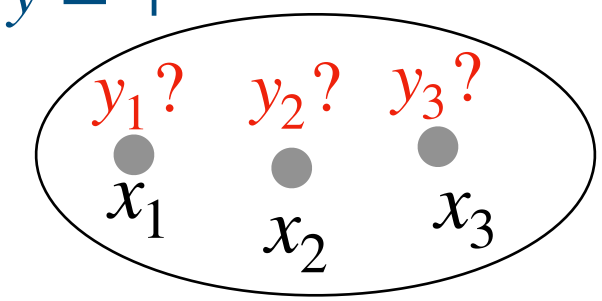
$y_i' \neq y_i$ $\quad y_i' = +$

$\quad\quad\quad\; y_i = -$

Inexact labels

coarser-grained labels: $(\{x_1, x_2, x_3\}, y)$

$y = +$



Domain heuristics

has_keyword(“happy”*)?*

$x_i \longrightarrow y_i = +$

has_keyword(“sad”*)?*

$x_i \longrightarrow y_i = -$

Weakly Supervised Learning (WSL)

- What is **weak** supervision?

Inaccurate labels

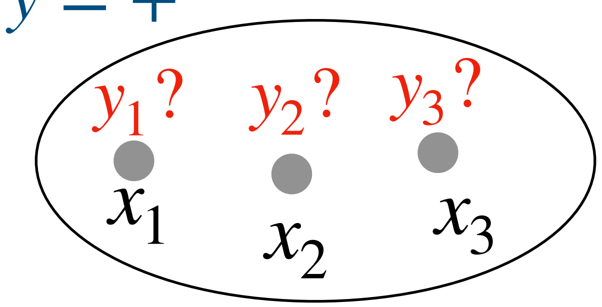
$$D_L = \{(x_i, y_i')\}_{i=1}^{N}$$

$y_i' \neq y_i$ $\quad$ $y_i' = +$ $\quad$ $y_i = -$

Inexact labels

coarser-grained labels: $(\{x_1, x_2, x_3\}, y)$

$y = +$



Domain heuristics

has_keyword(“happy”)?

$x_i \longrightarrow y_i = +$

has_keyword(“sad”)?

$x_i \longrightarrow y_i = -$

- Why leverage **weak** supervision?

Informative

correlate
with ground-truth

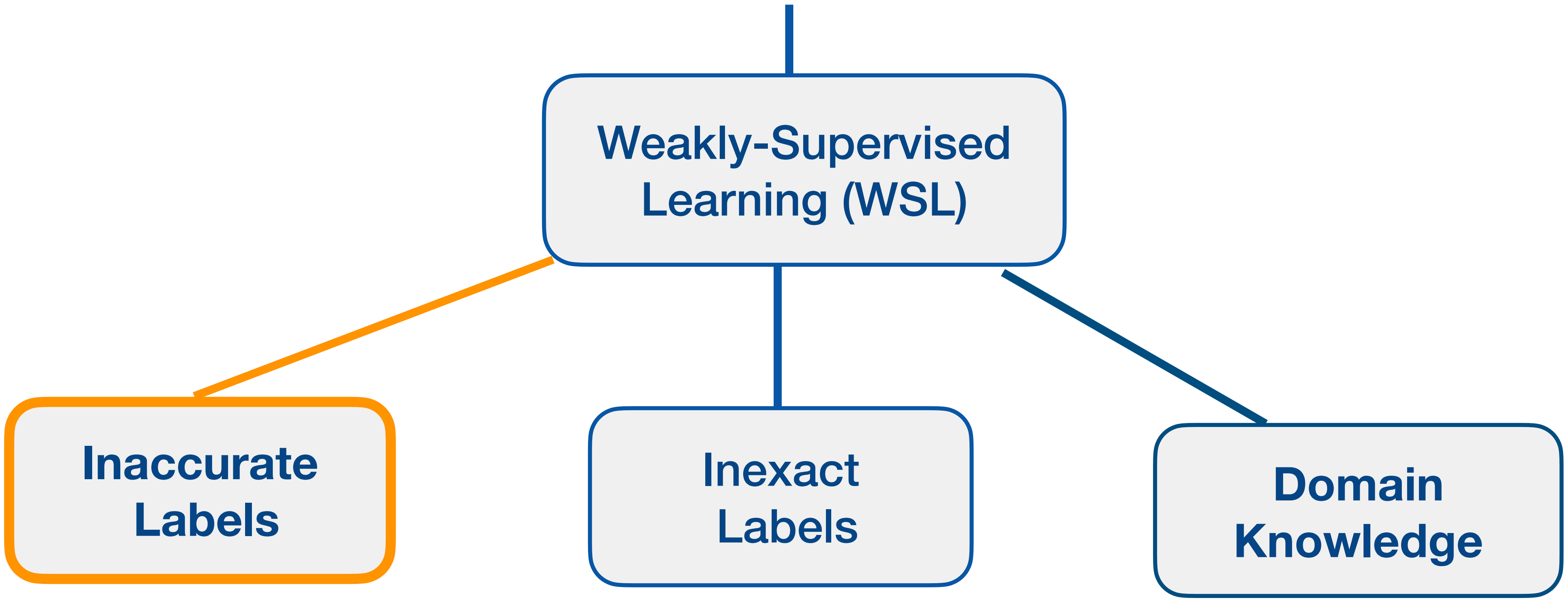
Cheap

abundant /
easy to collect

Scalable

can scale to huge
amounts of unlabeled data

WSL Taxonomy



WSL - Leveraging Inaccurate Labels

- **Inaccurate Labels:** observed label y_i' may differ from ground-truth label y_i

$$D_L = \{(x_i, y_i')\}_{i=1}^{N}$$

WSL - Leveraging Inaccurate Labels

- **Inaccurate Labels:** observed label y_i' may differ from ground-truth label y_i

$$D_L = \{(x_i, y_i')\}_{i=1}^{N}$$

- **Crowdsourcing** noisy labels**:**
  - **Redundancy trick**: get **multiple** noisy annotations per instance
  - Estimating ground-truth $\hat{y}$:
    - **majority voting** is effective & widely used
    - model **quality** of each individual **annotator** effective

[Sheng et al., 2008]

WSL - Leveraging Inaccurate Labels

- **Inaccurate Labels:** observed label y_i' may differ from ground-truth label y_i

$$D_L = \{(x_i, y_i')\}_{i=1}^{N}$$

- **Crowdsourcing** noisy labels**:**
  - **Redundancy trick**: get **multiple** noisy annotations per instance
  - Estimating ground-truth $\hat{y}$:
    - **majority voting** is effective & widely used
    - model **quality** of each individual **annotator** effective

[Sheng et al., 2008]

(-) expensive to achieve multiple labels per data point

Trade-off between **quantity** and **redundancy**

[Sheng et al., 2008]

WSL - Leveraging Inaccurate Labels

- **Inaccurate Labels:** observed label y'_i may differ from ground-truth label y_i

$$D_L = \{(x_i, y'_i)\}_{i=1}^{N}$$

- **Crowdsourcing** noisy labels**:**
  - **Redundancy trick**: get **multiple** noisy annotations per instance
  - Estimating ground-truth $\hat{y}$:
    - **majority voting** is effective & widely used
    - model **quality** of each individual **annotator** effective

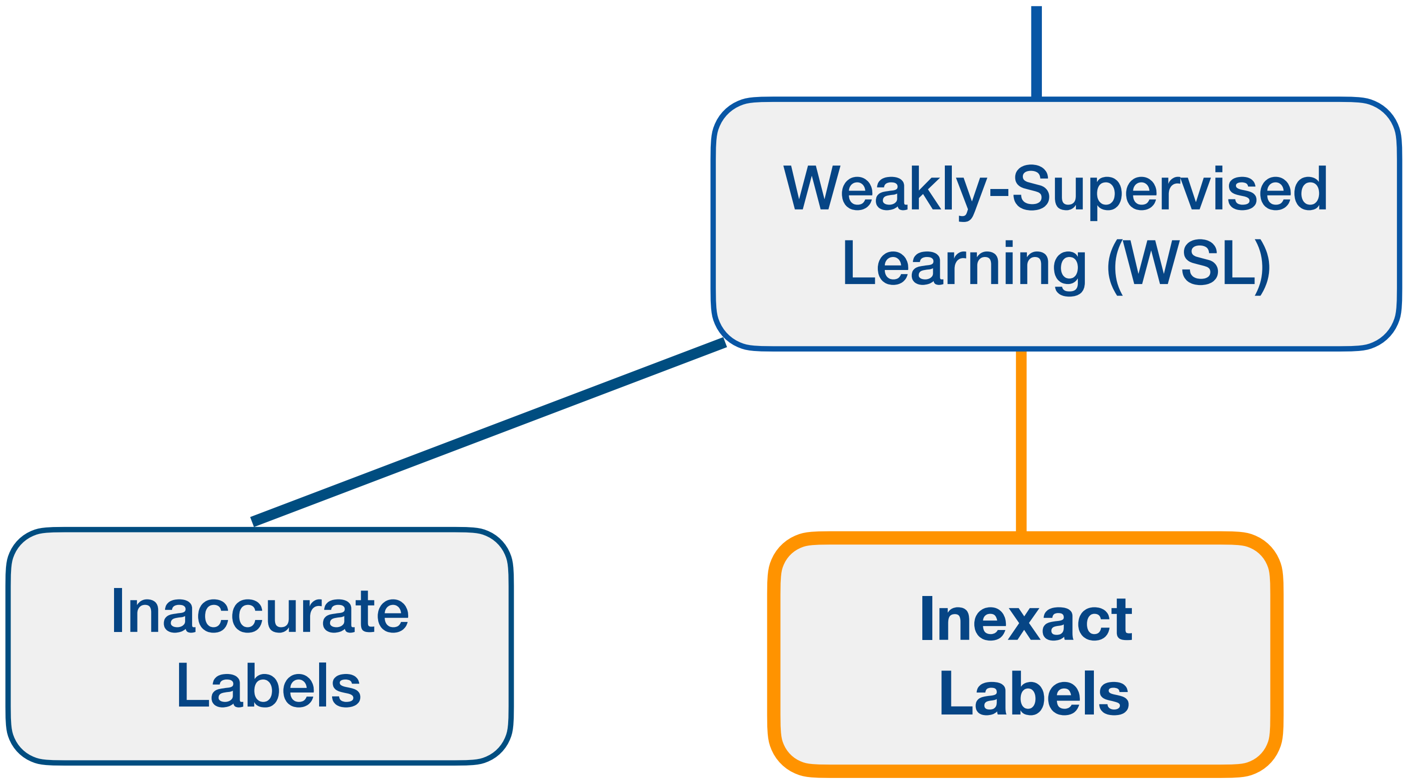
[Sheng et al., 2008]

  (-) expensive to achieve multiple labels per data point

- **Learning** with noisy labels: single label per instance
  - random classification noise: y_i has been flipped to y'_i with probability p_i
  - need assumptions about noise structure:
    - class-conditional noise: $p_i = P(y'_i \mid y_i, x_i) = P(y'_i \mid y_i)$

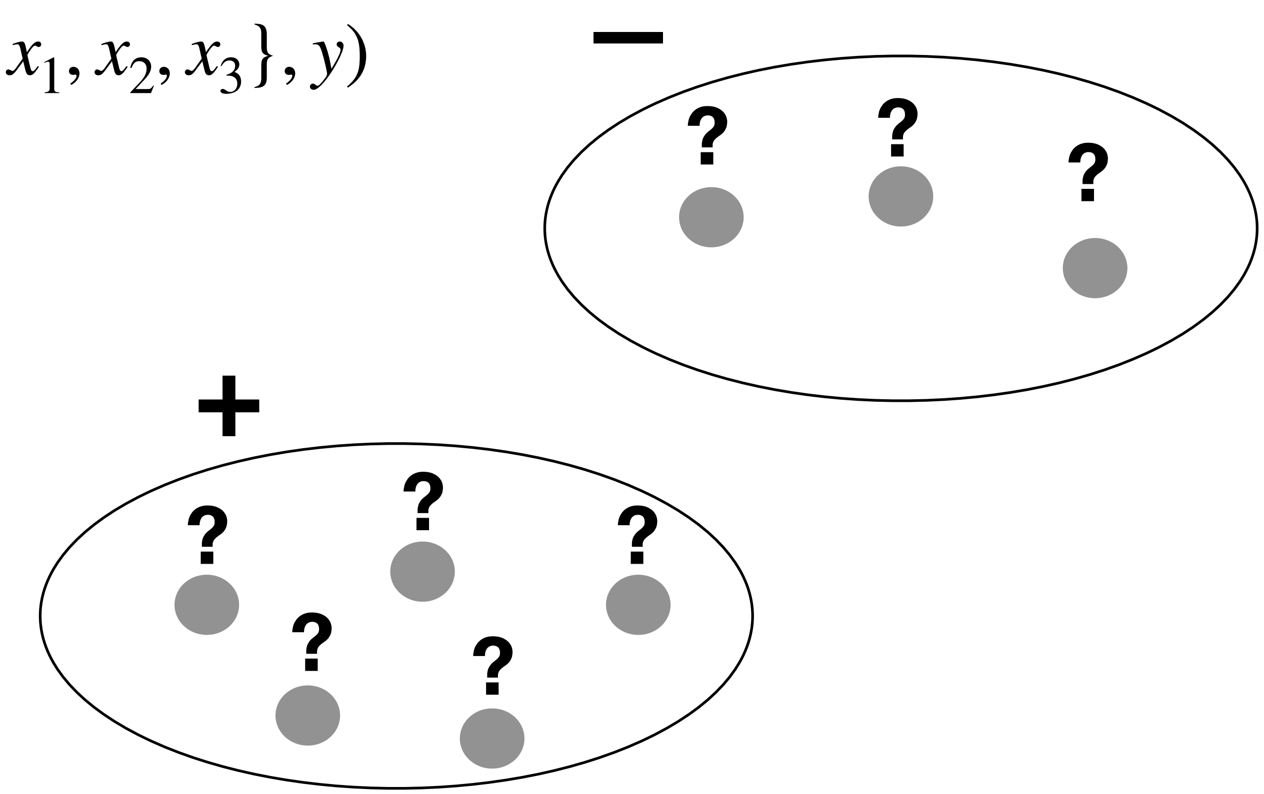
[Natarajan et al., 2013]

WSL Taxonomy



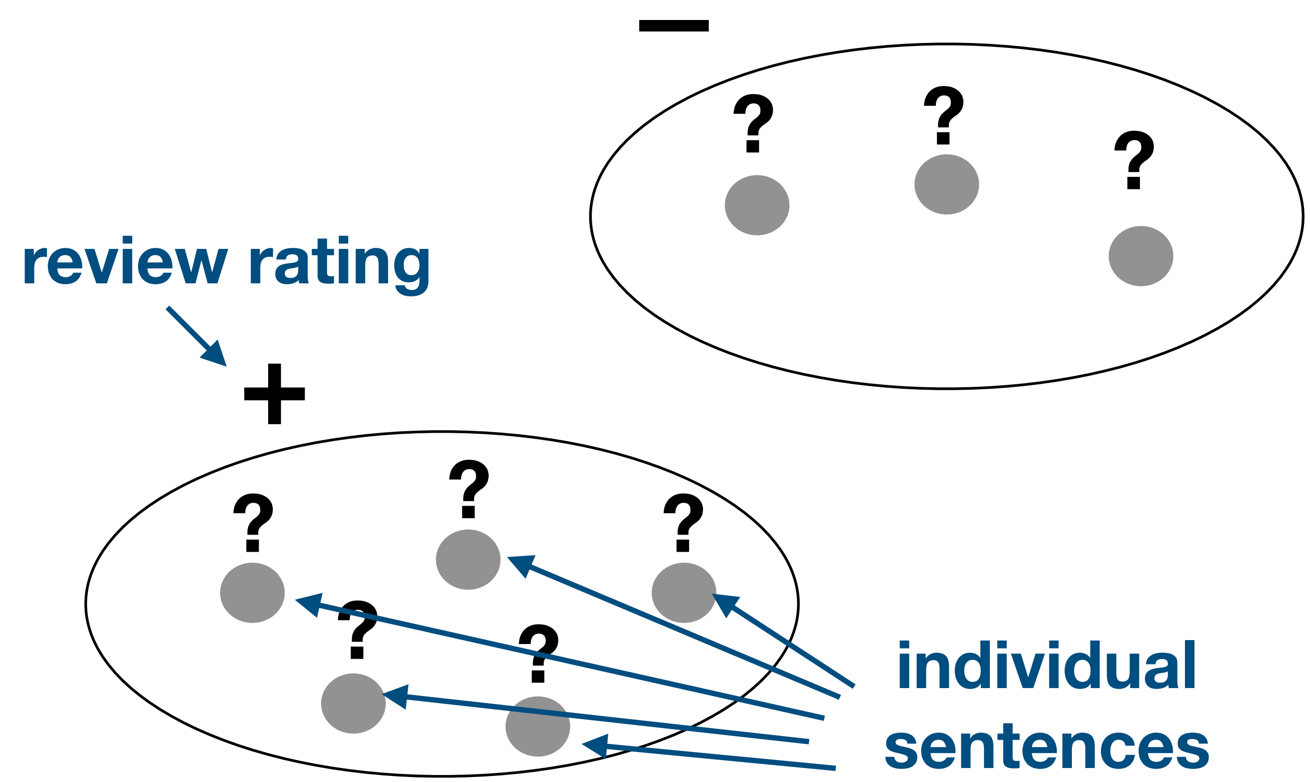
WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels $(\{x_1, x_2, x_3\}, y)$
    - "Bags of instances"
    - **Observed** bag labels y
    - **Unobserved** instance labels y_i



WSL - Leveraging Inexact Labels

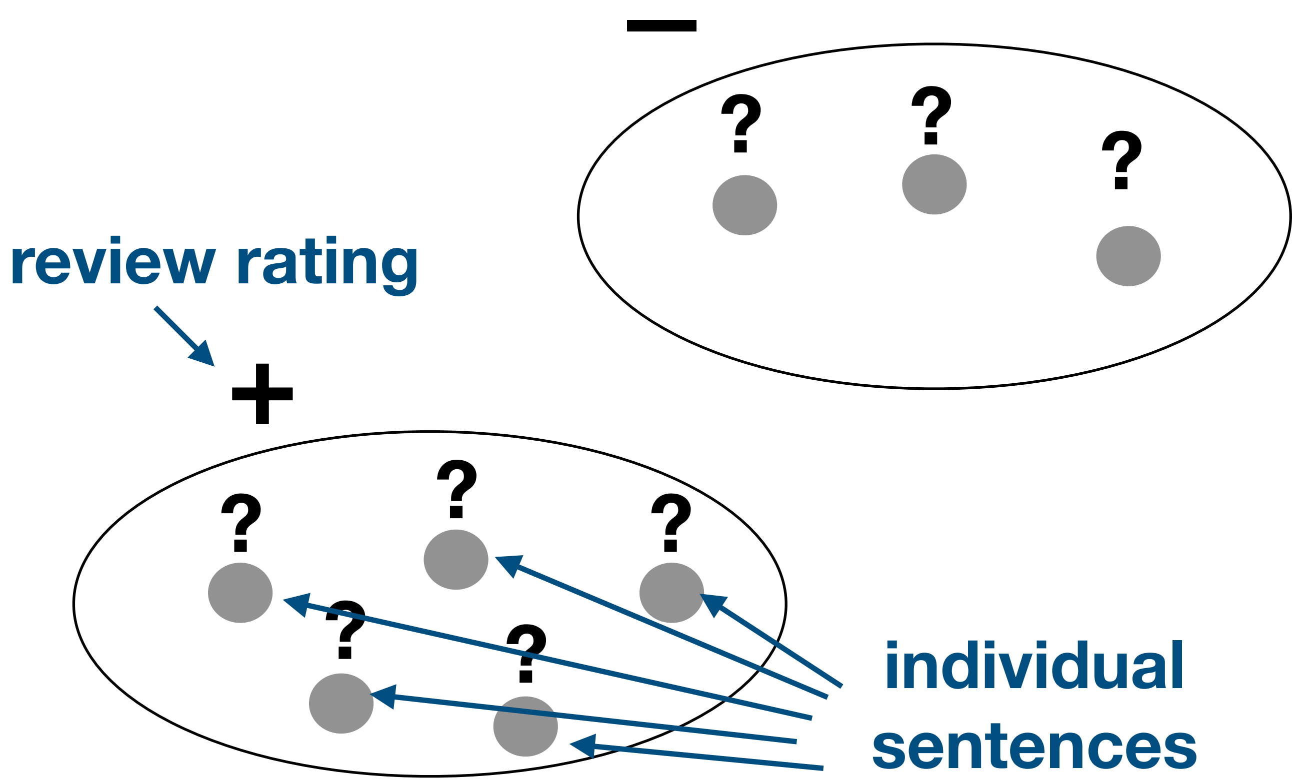
- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification



WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \; \forall i = 1..T$

  (-) introduces noisy labels



e.g., **rating = +**

$x_1 =$ "I loved the food", $y_1 = +$

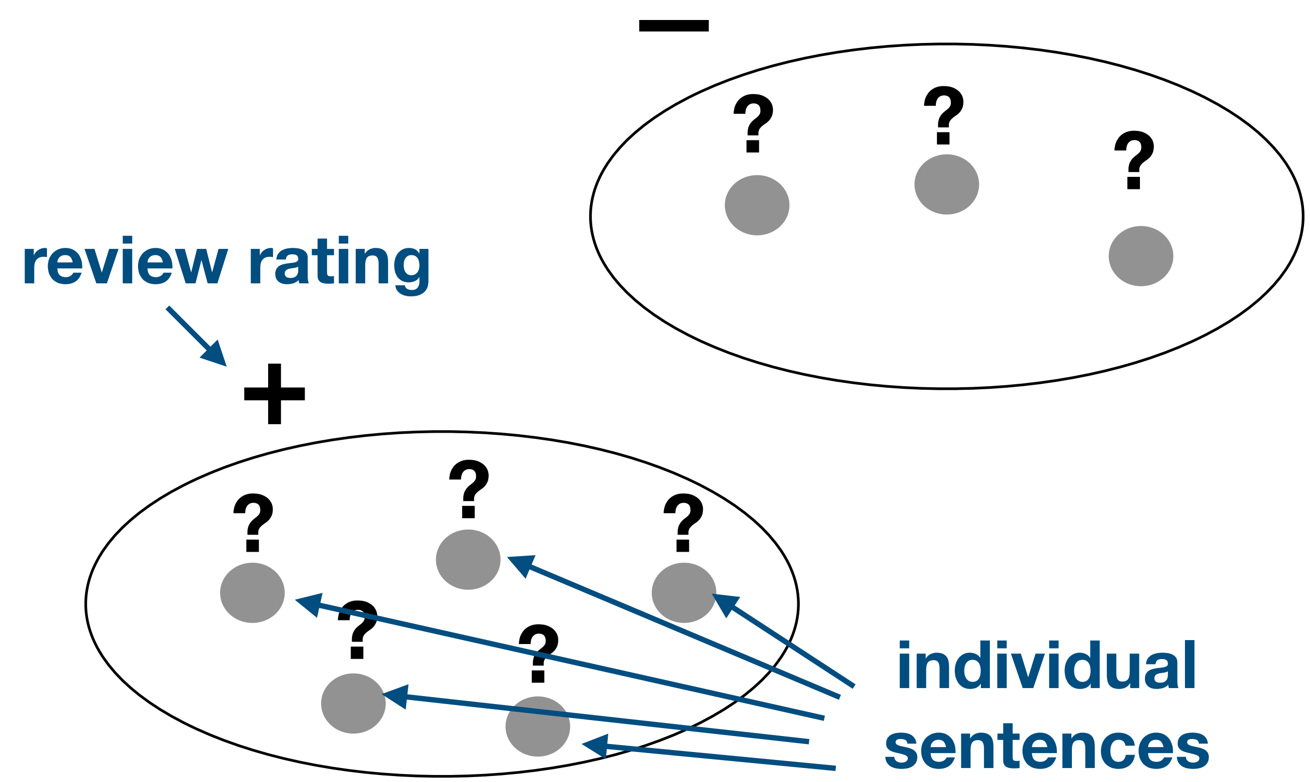
$x_2 =$ "The service was bad", $y_2 = -$

$x_3 =$ "Overall I liked it", $y_3 = +$

WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \ \forall i = 1..T$

  (-) introduces noisy labels



- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption:

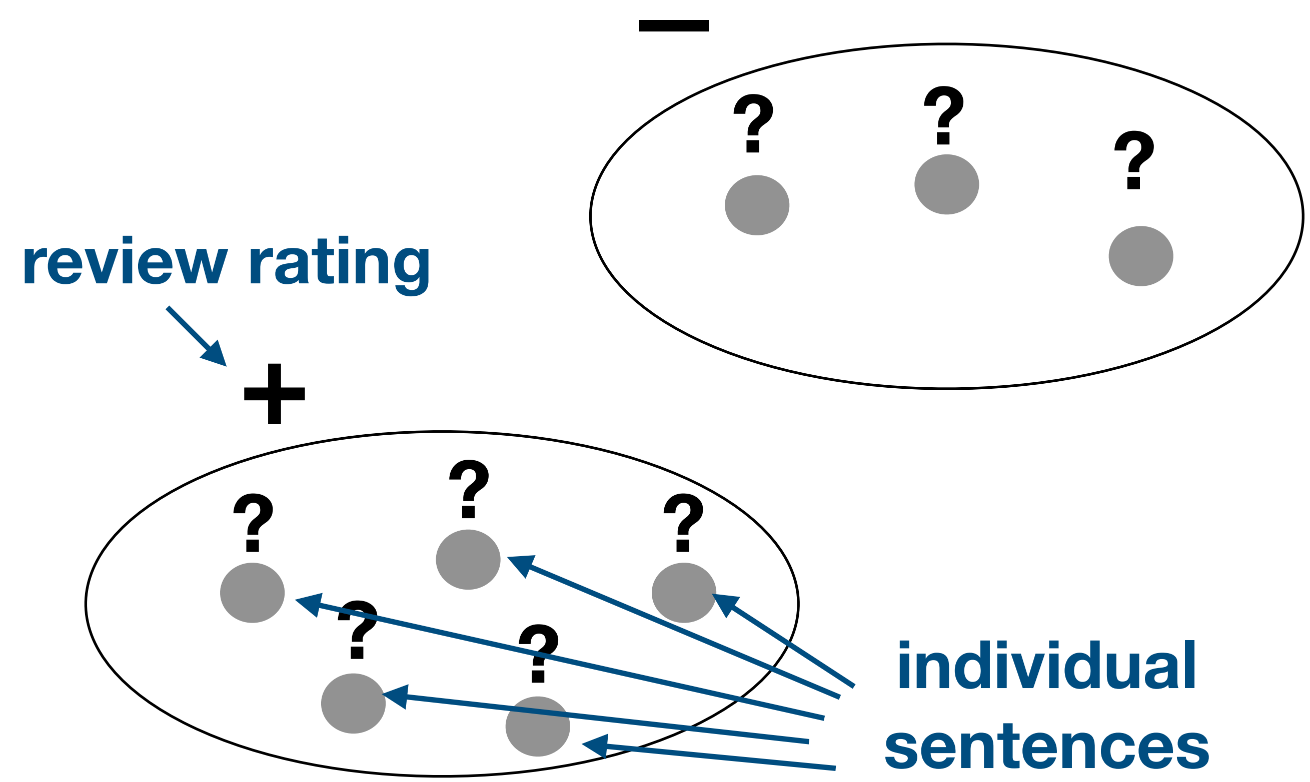
[Andrews et al., 2002]

$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently:} \quad y = \max(y_1, \ldots, y_T)\,)$$

WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \;\forall i = 1..T$

  (-) introduces noisy labels



- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption:

[Andrews et al., 2002]

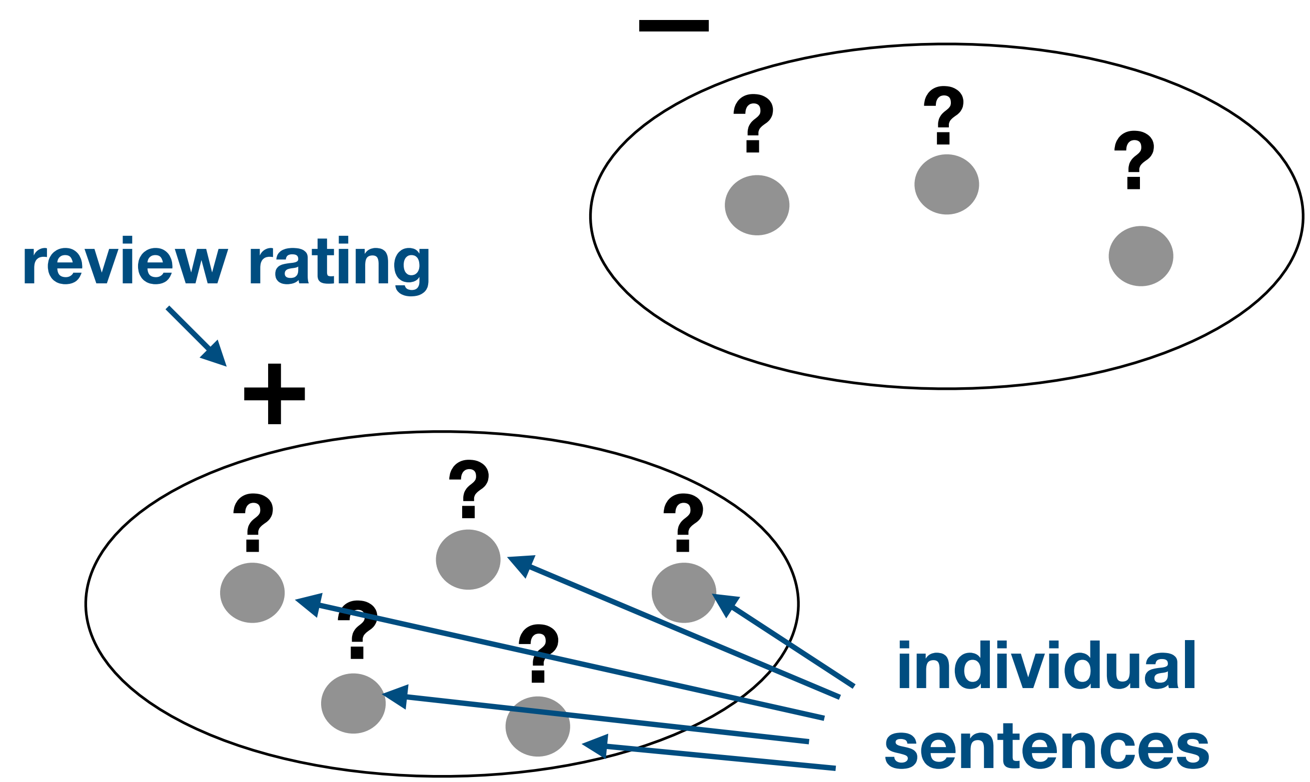
$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently: } \quad y = \max(y_1, \ldots, y_T)\text{)}$$

(-) does not always hold true in text classification

WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \; \forall i = 1..T$

  (-) introduces noisy labels



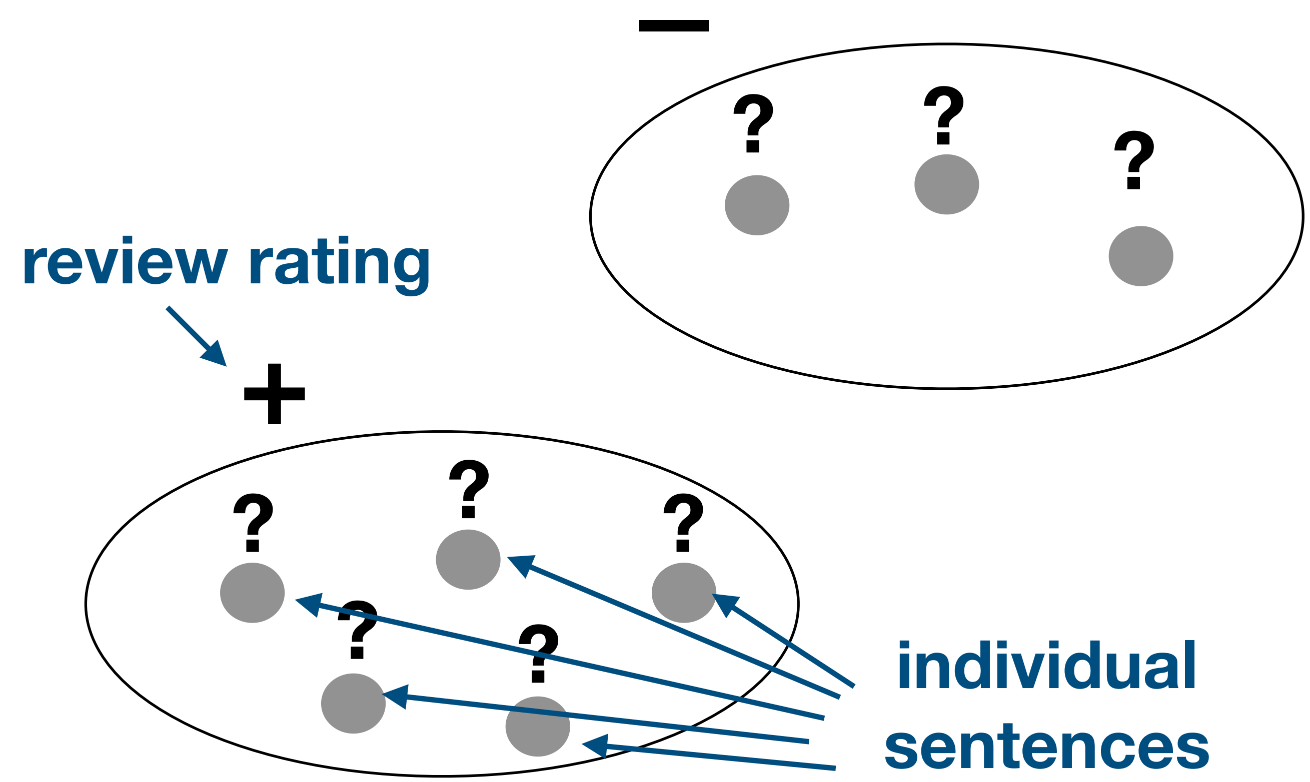
- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption: [Andrews et al., 2002]

$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently:} \quad y = \max(y_1, \ldots, y_T)\,)$$

  - **More natural** assumptions: [Kotzias et al., 2015]
    - average: $$y = \frac{1}{T} \sum_i y_i$$

WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i



- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \ \forall i = 1..T$
  - **(-) introduces noisy labels**

- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption:

[Andrews et al., 2002]

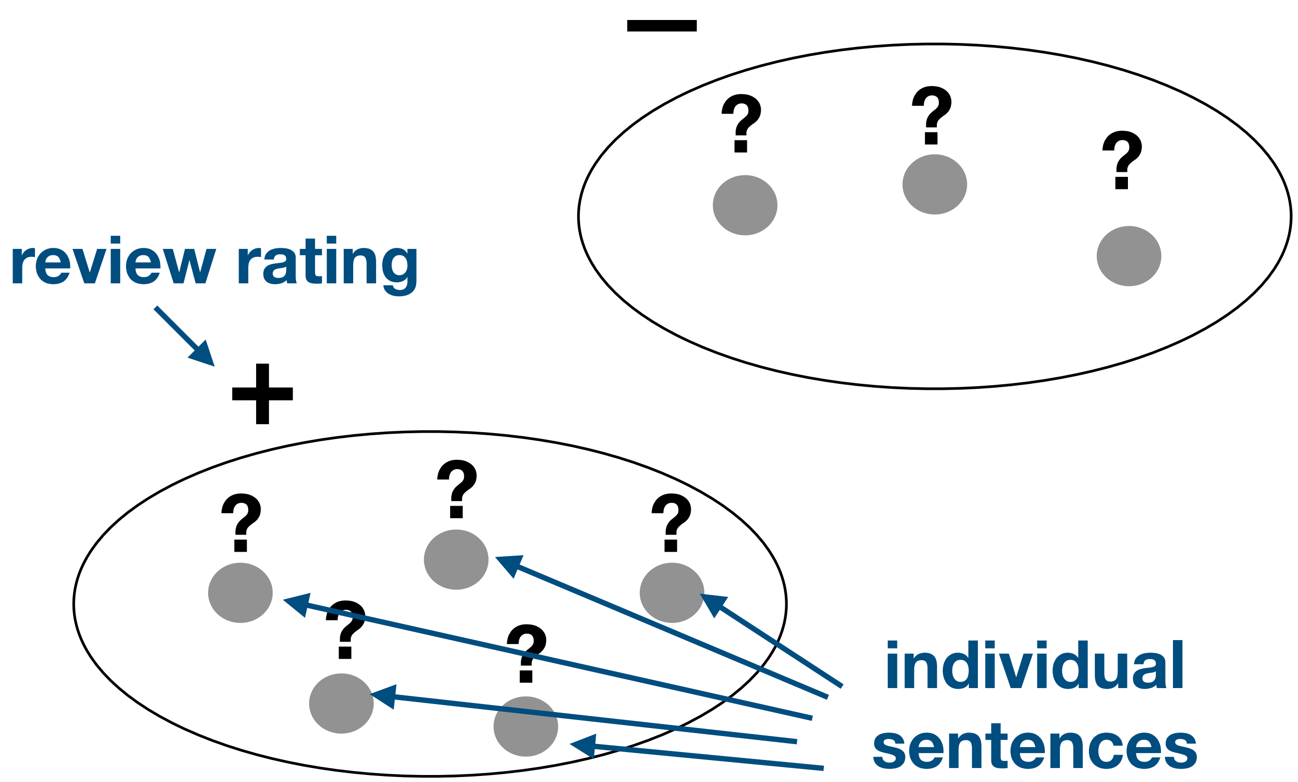
$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently:} \quad y = \max(y_1, \ldots, y_T)\text{)}$$

  - **More natural** assumptions:
    - average: $y = \frac{1}{T} \sum_i y_i$

[Kotzias et al., 2015]

(-) ignores the relative importance of instances

WSL - Leveraging Inexact Labels



- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \; \forall i = 1..T$

  (-) introduces noisy labels

- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption:

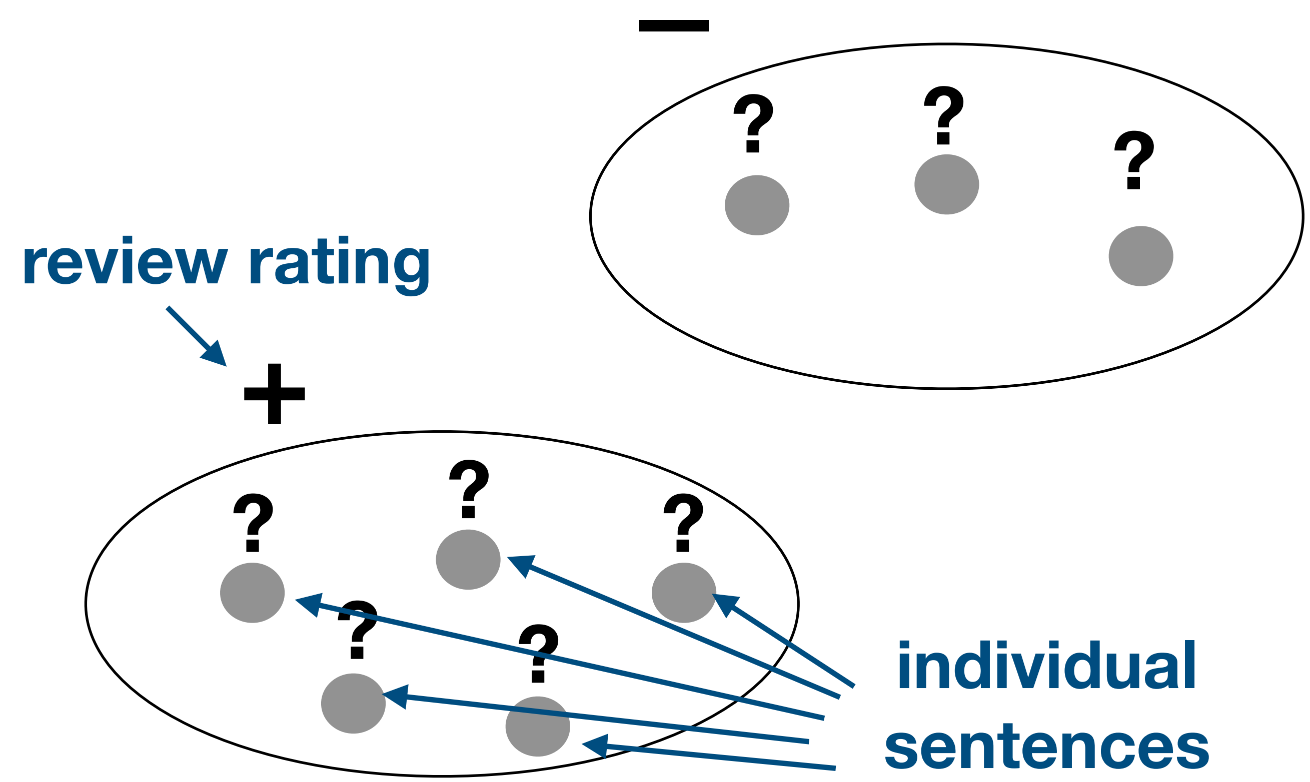
[Andrews et al., 2002]

$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently:} \quad y = \max(y_1, \ldots, y_T)\,)$$

- **More natural** assumptions:
  - average: $y = \frac{1}{T} \sum_i y_i$ [Kotzias et al., 2015]
  - **weighted** average: $y = \frac{1}{T} \sum_i \alpha_i y_i$ [Angelidis & Lapata, 2018]

WSL - Leveraging Inexact Labels

- **Inexact Labels:** coarser-grained labels
  - "Bags of instances"
  - **Observed** bag labels y
  - **Unobserved** instance labels y_i
- **Example:** review sentiment classification
- **Naive approach:** $y_i = y \ \forall i = 1..T$
  - (-) introduces noisy labels



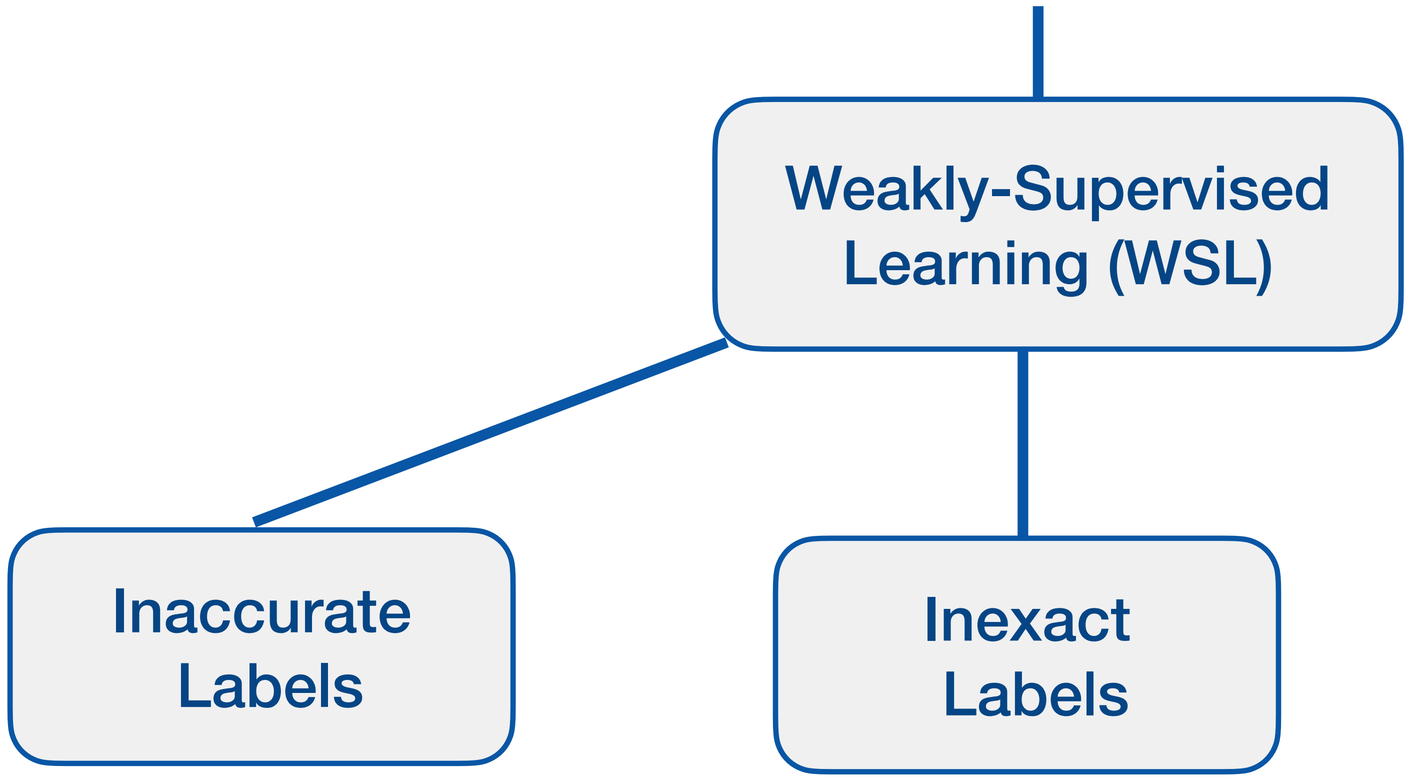
- **Multiple Instance Learning (MIL):** $y = \mathrm{AGG}(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{T})$
  - "at least one" assumption: [Andrews et al., 2002]

$$y = + \quad \Leftrightarrow \quad \exists y_i : y_i = + \quad \text{(equivalently:} \quad y = \max(y_1, \ldots, y_T)\text{)}$$

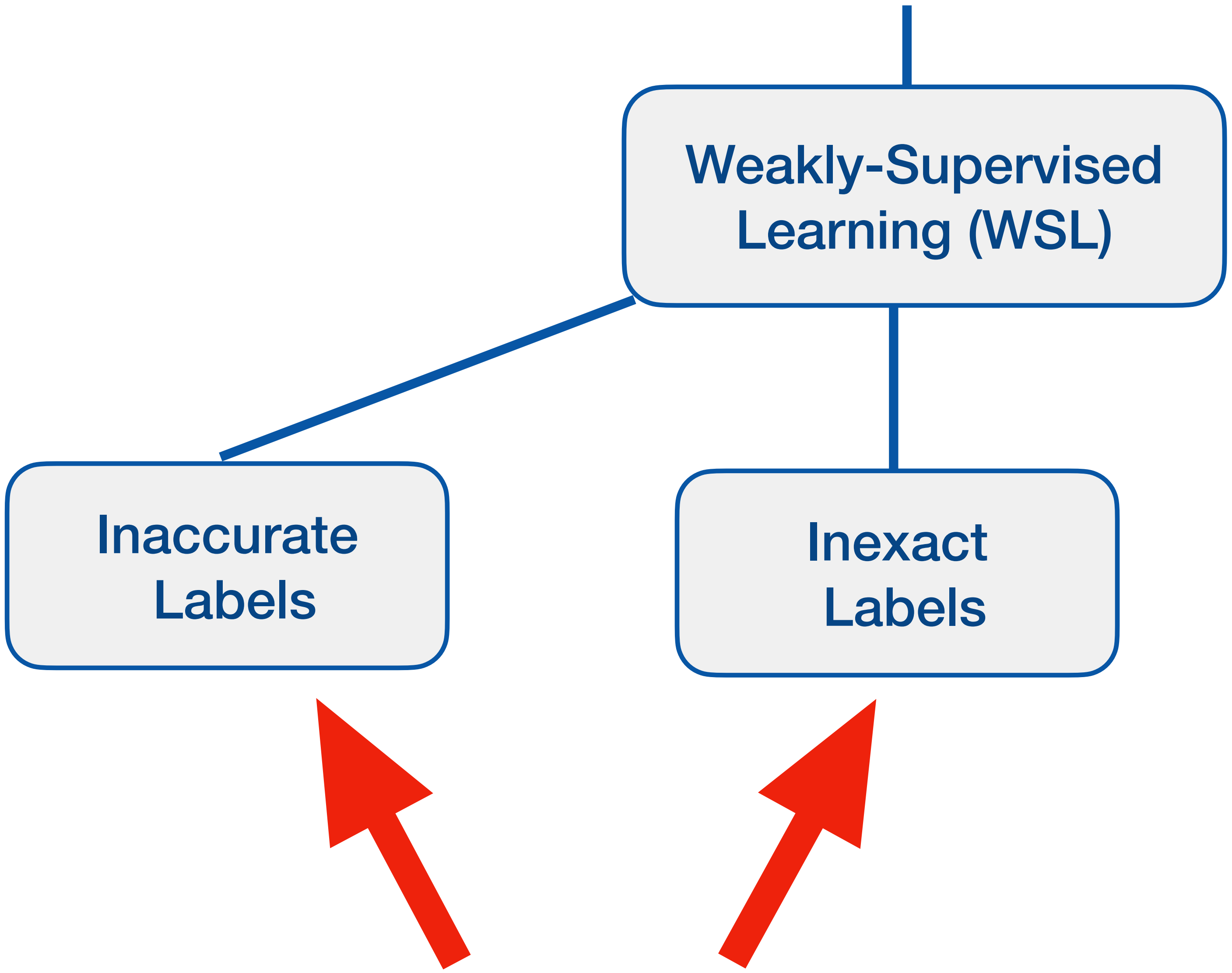
- **More natural** assumptions:
  - average: $y = \frac{1}{T} \sum_i y_i$ [Kotzias et al., 2015]
  - **weighted** average: $y = \frac{1}{T} \sum_i \alpha_i y_i$ [Angelidis & Lapata, 2018]

also learned!

Weakly Supervised Learning (WSL)



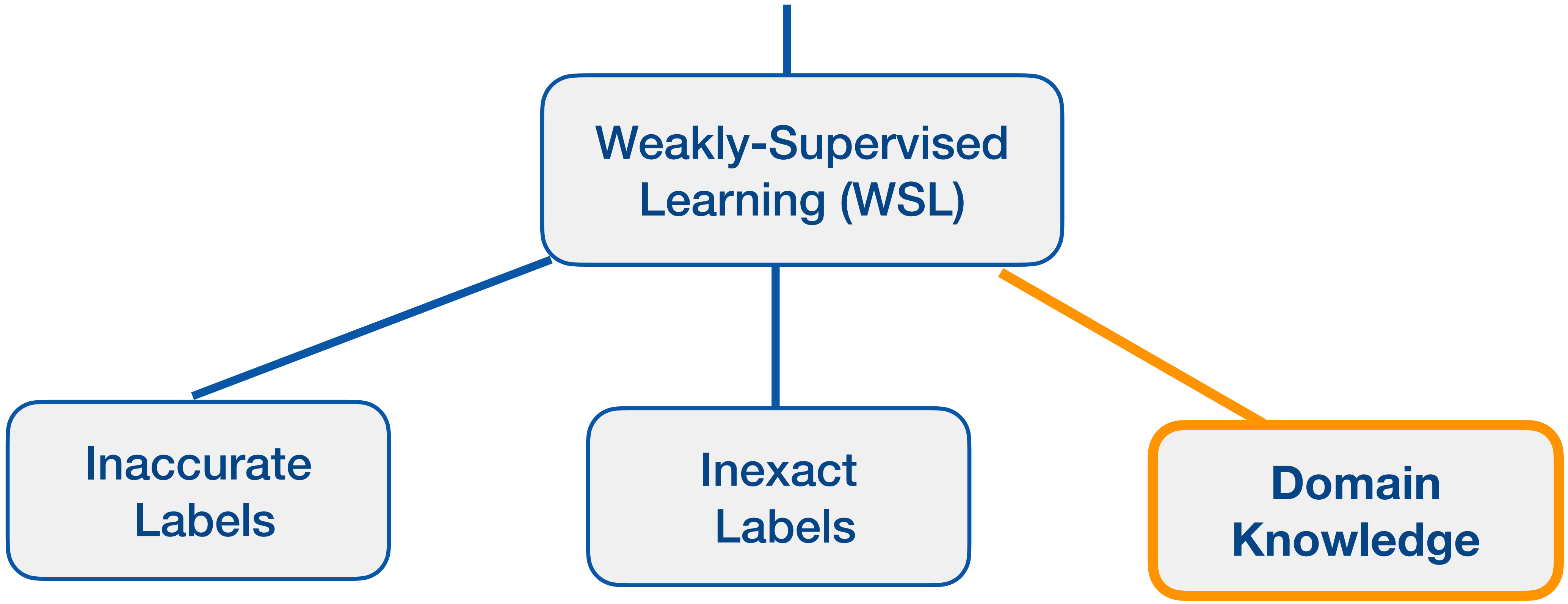
Weakly Supervised Learning (WSL)



(-) restricted:

- Only support “**coarse**” assumptions
- Same assumptions **regardless** domain
- **Worst case:** what if NO training labels are available?

Weakly Supervised Learning (WSL)



[Yarowsky, 1995]
[Riloff & Jones, 1999]
[Collins & Singer, 1999]
[Agichtein & Gravano, 2000]
[Ganchev et al., 2010]
[Ratner et al., 2017]

- **Focus:** Leveraging domain knowledge as heuristics for weak supervision

What is “Domain Knowledge”?

- **What is domain knowledge in our setting?**
  - Prior expert knowledge about the specific domain/task
  - **Differrent** knowledge for different tasks

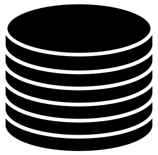
Sentiment Classification

≠

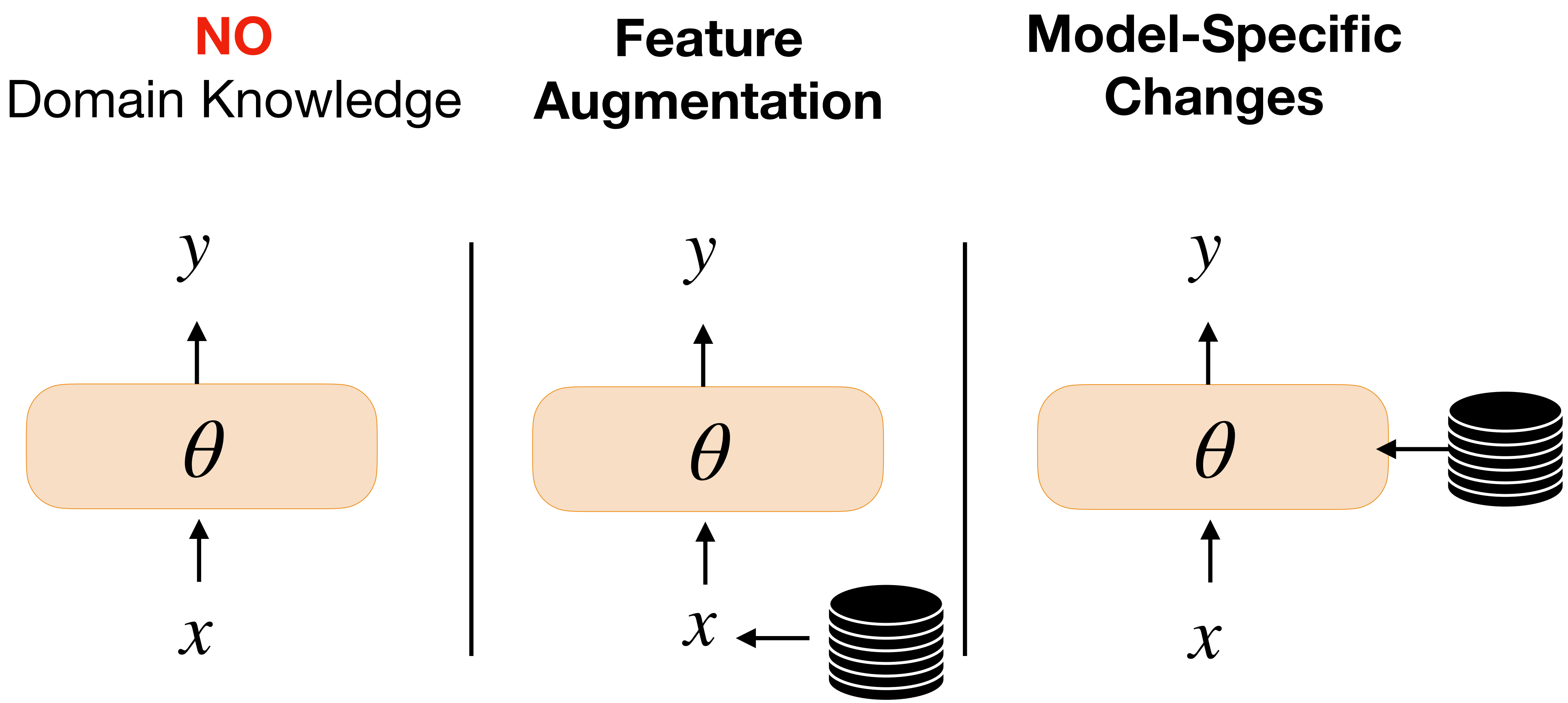
News Topics Classification

≠

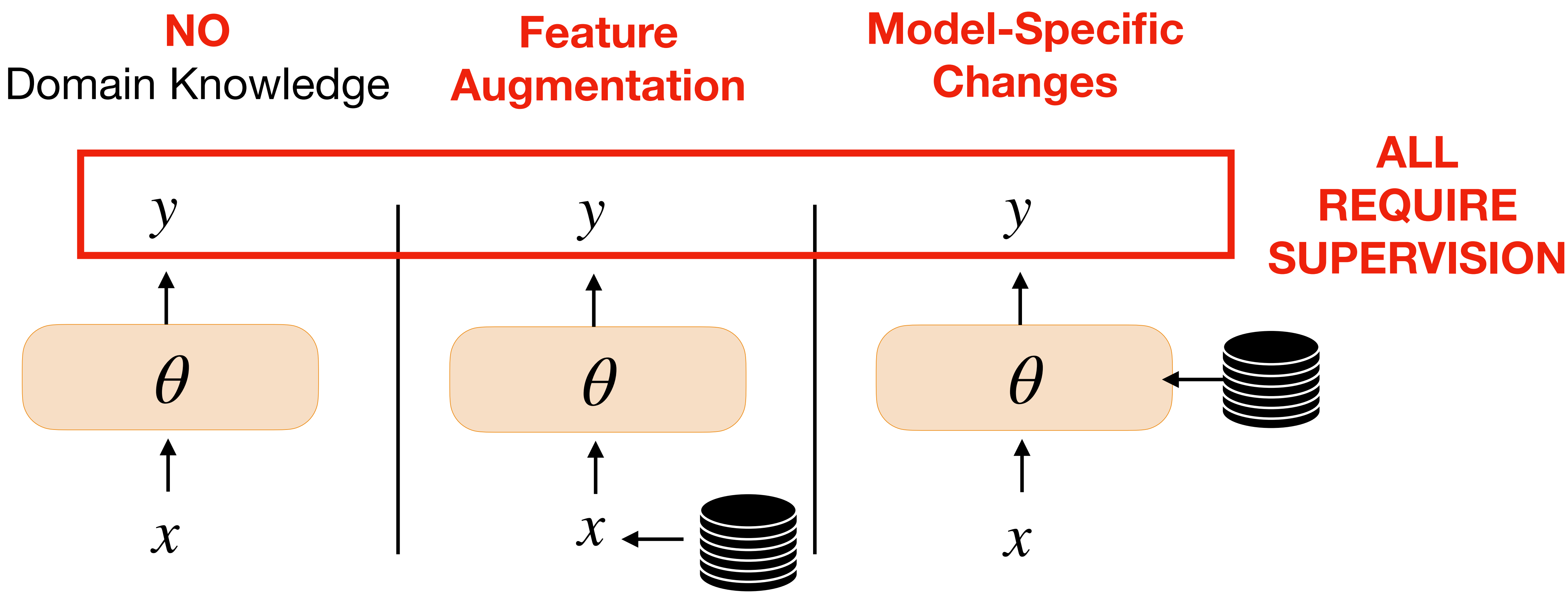
Emergency Events Detection

- **Examples of domain knowledge :**
  - **Domain-specific lexicons:**
    - e.g., {‘*angry*’: -0.8, ‘*happy*’: 0.7, ‘*of*’: 0.0, …}
  - **Heuristic rules for each target class:**
    - *e.g., has_keyword(“happy”) ->* **positive** sentiment
    - e.g., has_keyword(“*money*”) -> **price** topic
    - e.g., has_emoji() -> **positive** sentiment
  - **Expert-curated knowledge base:** 

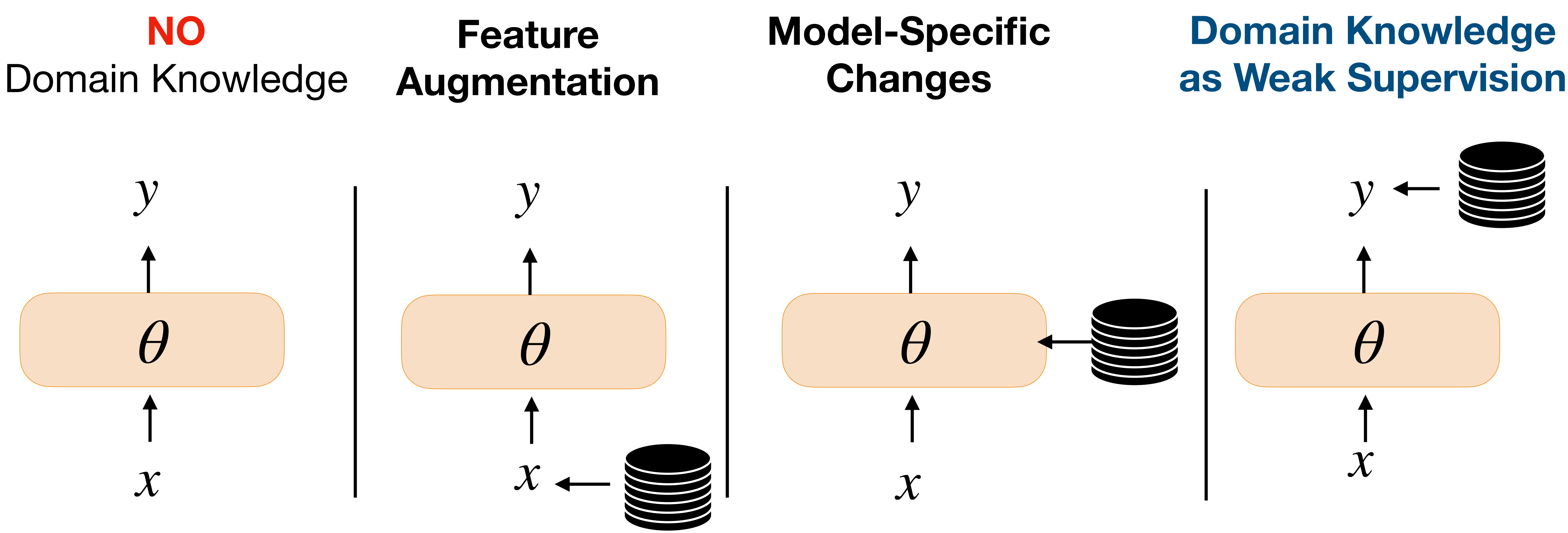
How to Leverage “Domain Knowledge”?



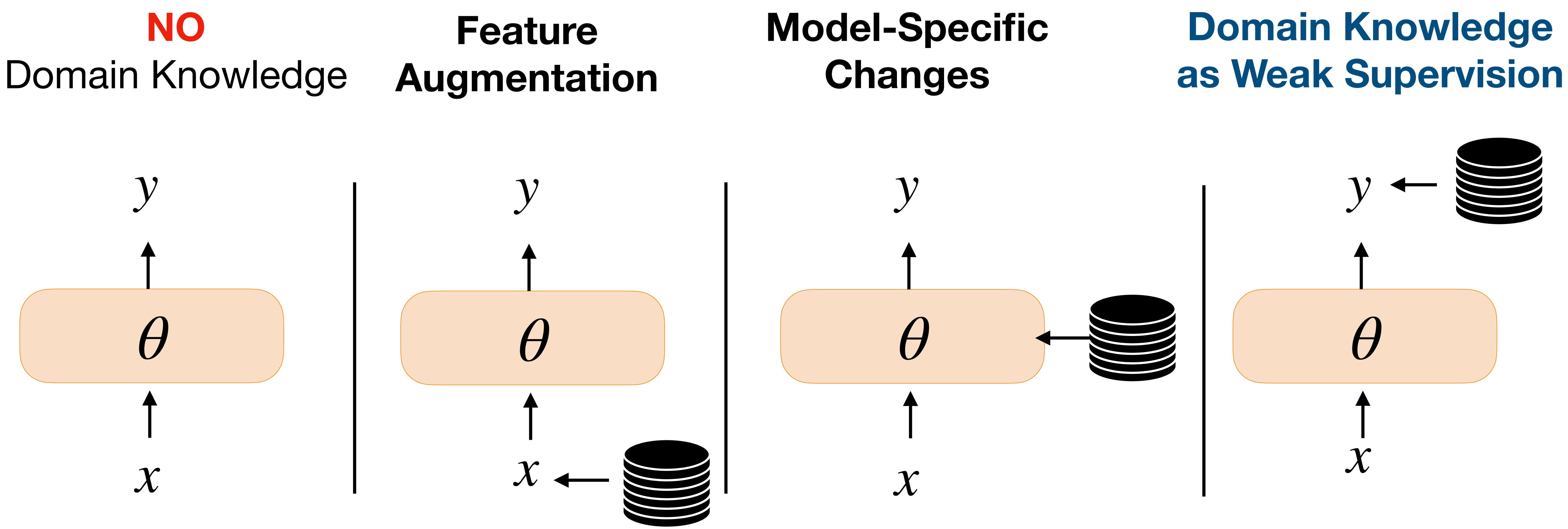
How to Leverage "Domain Knowledge"?



How to Leverage “Domain Knowledge”?



How to Leverage “Domain Knowledge”?



- Our focus: leveraging domain knowledge as **weak supervision**
  - e.g., to create **more labels**
  - e.g., to create **regularizers**

Leveraging Domain Knowledge as Weak Supervision

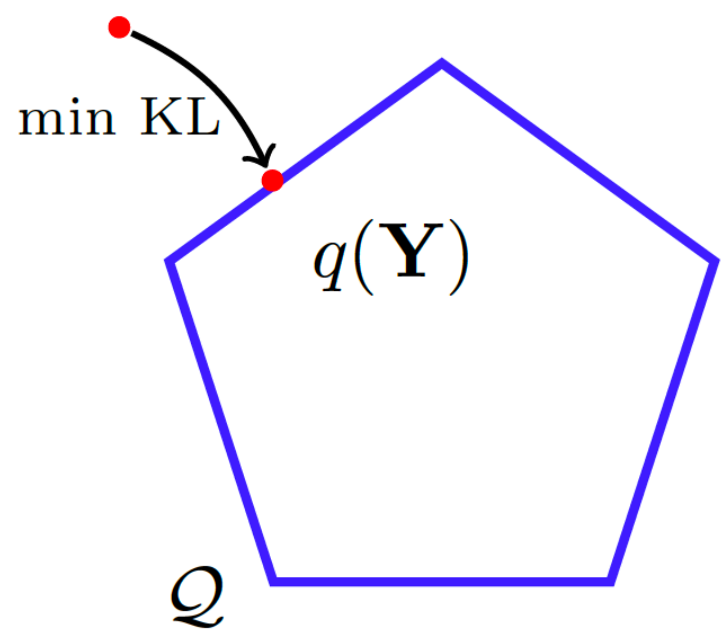
- **Posterior regularization (PR):**

[Ganchev et al., 2010]

  - Use domain heuristics to create linear **constraints** $\mathcal{Q}$ …
  - … on **posterior** distributions of latent variable models $p_\theta(\mathbf{Y}|\mathbf{X})$
  - Constraints hold **in expectation**
  - Examples:

    Classification: Positive class should be predicted 75%

    POS tagging: There should be at least one "VERB" in y



Leveraging Domain Knowledge as Weak Supervision

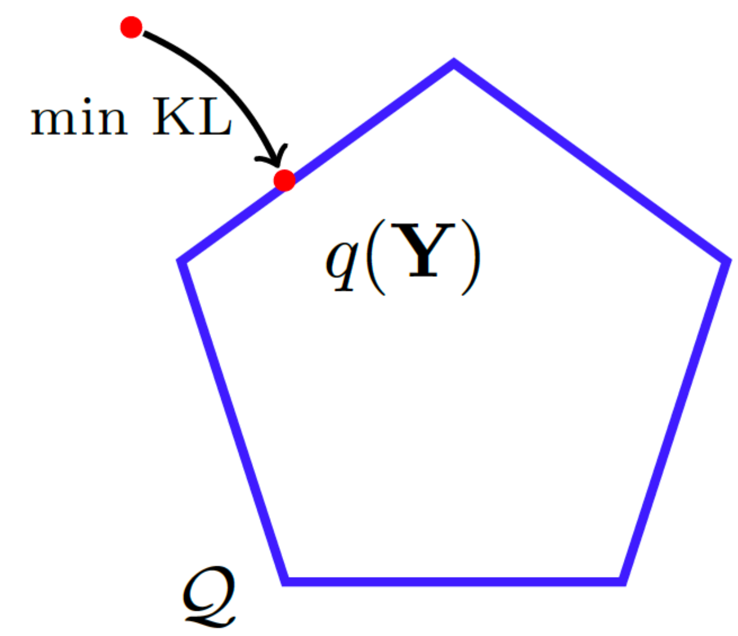
[Ganchev et al., 2010]

- **Posterior regularization (PR):**
  - Use domain heuristics to create linear **constraints** $\mathcal{Q}$ …
  - … on **posterior** distributions of latent variable models $p_\theta(\mathbf{Y}|\mathbf{X})$
  - Constraints hold **in expectation**
  - Examples:

    Classification: Positive class should be predicted 75%

    POS tagging: There should be at least one "VERB" in y

  (-) limited expressiveness



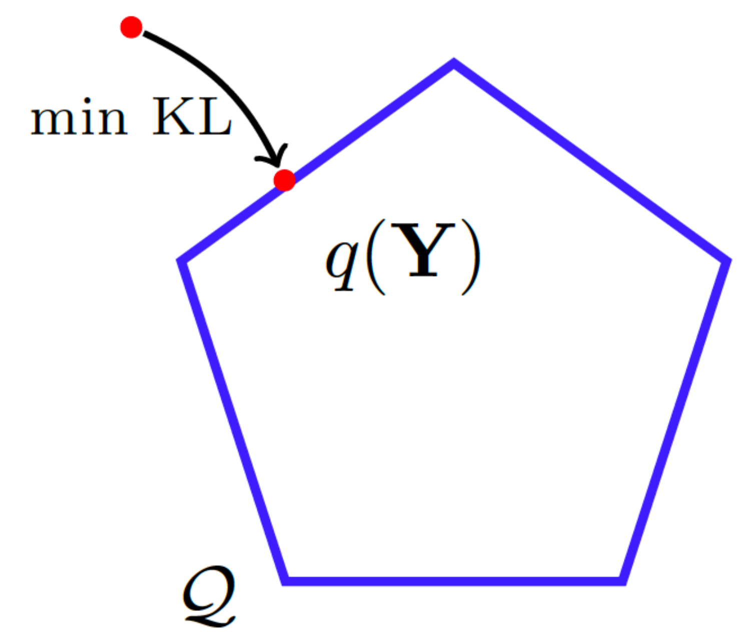
Leveraging Domain Knowledge as Weak Supervision

- **Posterior regularization (PR):** [Ganchev et al., 2010]
  - Use domain heuristics to create linear **constraints** $\mathcal{Q}$ …
  - … on **posterior** distributions of latent variable models $p_\theta(\mathbf{Y}|\mathbf{X})$
  - Constraints hold **in expectation**
  - Examples:

    Classification: Positive class should be predicted 75%

    POS tagging: There should be at least one "VERB" in y

  (-) limited expressiveness



- **Data programming (DP):**
  - Leverage heuristics as **instance-level** labeling functions (LFs) [Ratner et al., 2017]

  (+) expressiveness

```
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
```

Leveraging Domain Knowledge as Weak Supervision

- **Posterior regularization (PR):**

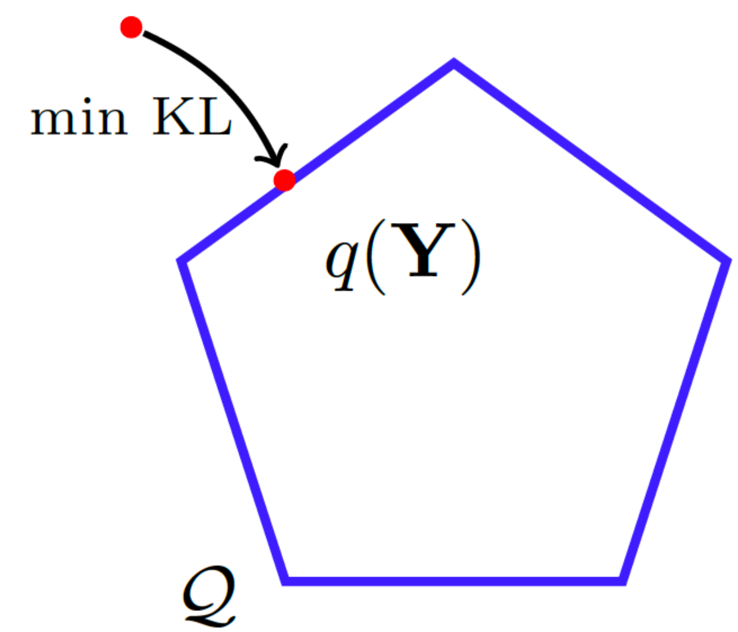
[Ganchev et al., 2010]

  - Use domain heuristics to create linear **constraints** $\mathcal{Q}$ …
  - … on **posterior** distributions of latent variable models $p_\theta(\mathbf{Y}|\mathbf{X})$
  - Constraints hold **in expectation**
  - Examples:

    Classification: Positive class should be predicted 75%

    POS tagging: There should be at least one "VERB" in y

  (-) limited expressiveness



- **Data programming (DP):**
  - Leverage heuristics as **instance-level** labeling functions (LFs) [Ratner et al., 2017]

  (+) expressiveness

```
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
```

(-) PR and DP require a **sufficiently large** set of heuristics to effectively guide learning…

Collecting a sufficiently large set (lexicon / rules / KB)
may be **expensive**

How to leverage
a small **seed** set S (of words / rules / tuples)?

Leveraging Minimal Domain Knowledge via Bootstrapping

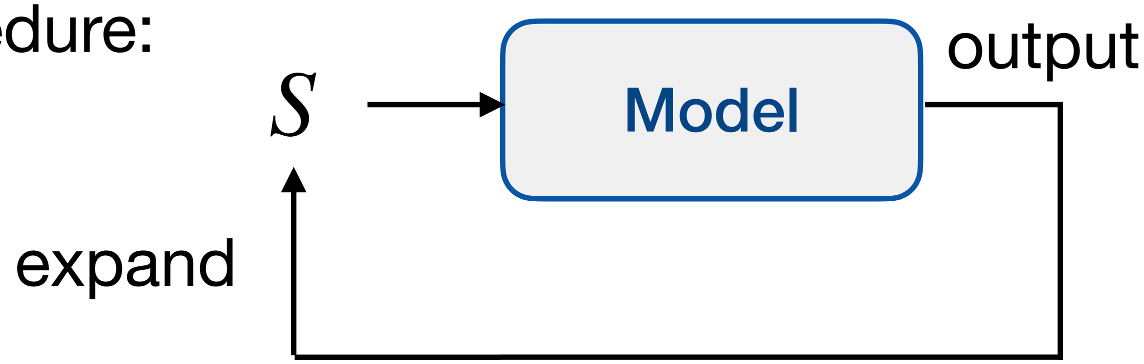
- **Challenge:** Seed set S has **limited** coverage (#datapoints where S applies)

Leveraging Minimal Domain Knowledge via Bootstrapping

- **Challenge:** Seed set S has **limited** coverage (#datapoints where S applies)
- **Bootstrapping algorithm**

[Yarowsky, 1995]

  - Increase coverage without extra supervision!
  - Iterative procedure:

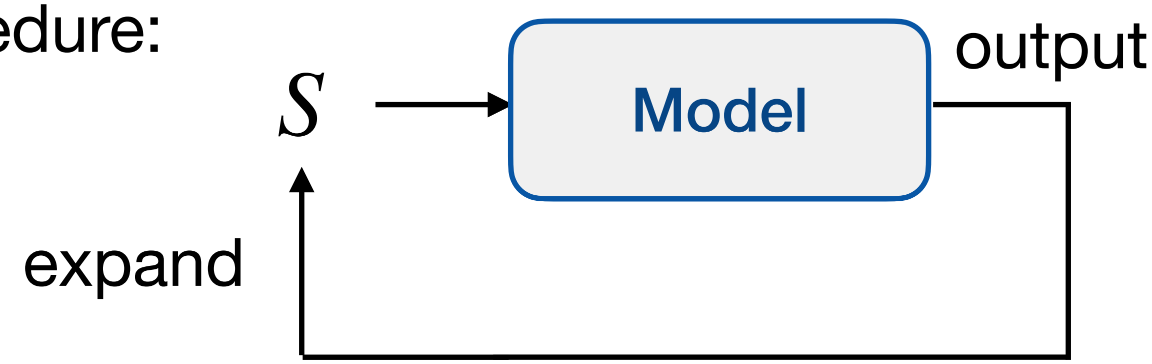


Leveraging Minimal Domain Knowledge via Bootstrapping

- **Challenge:** Seed set S has **limited** coverage (#datapoints where S applies)
- **Bootstrapping algorithm**

[Yarowsky, 1995]

  - Increase coverage without extra supervision!
  - Iterative procedure:



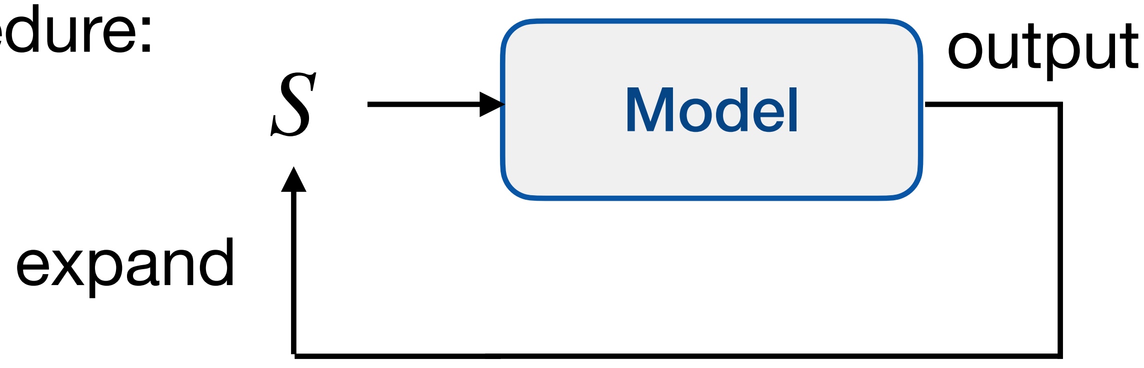
- Many successful applications of bootstrapping!
  - Seed **words** for information extraction [Riloff & Jones, 1999]
  - Seed **rules** for named entity recognition [Collins & Singer, 1999]
  - Seed **tuples** for relation extraction [Agichtein & Gravano, 2000]

Leveraging Minimal Domain Knowledge via Bootstrapping

- **Challenge:** Seed set S has **limited** coverage (#datapoints where S applies)
- **Bootstrapping algorithm**

[Yarowsky, 1995]

  - Increase coverage without extra supervision!
  - Iterative procedure:

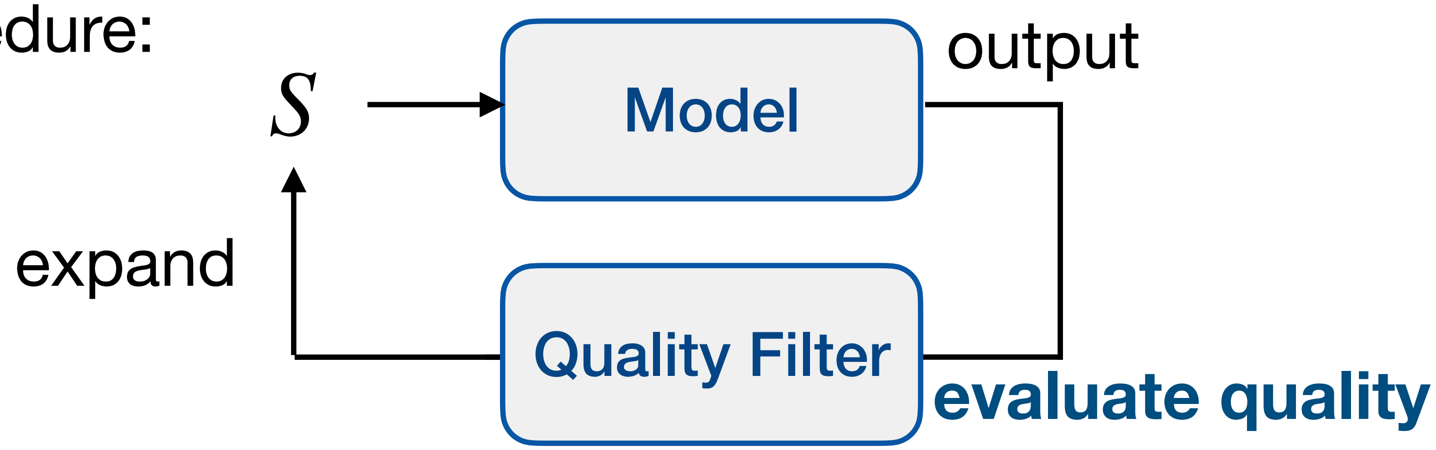


- Many successful applications of bootstrapping!
  - Seed **words** for information extraction [Riloff & Jones, 1999]
  - Seed **rules** for named entity recognition [Collins & Singer, 1999]
  - Seed **tuples** for relation extraction [Agichtein & Gravano, 2000]

- **Warning:** Model is unable to correct its own errors

Leveraging Minimal Domain Knowledge via Bootstrapping

- **Challenge:** Seed set S has **limited** coverage (#datapoints where S applies)
- **Bootstrapping algorithm** [Yarowsky, 1995]
  - Increase coverage without extra supervision!
  - Iterative procedure:

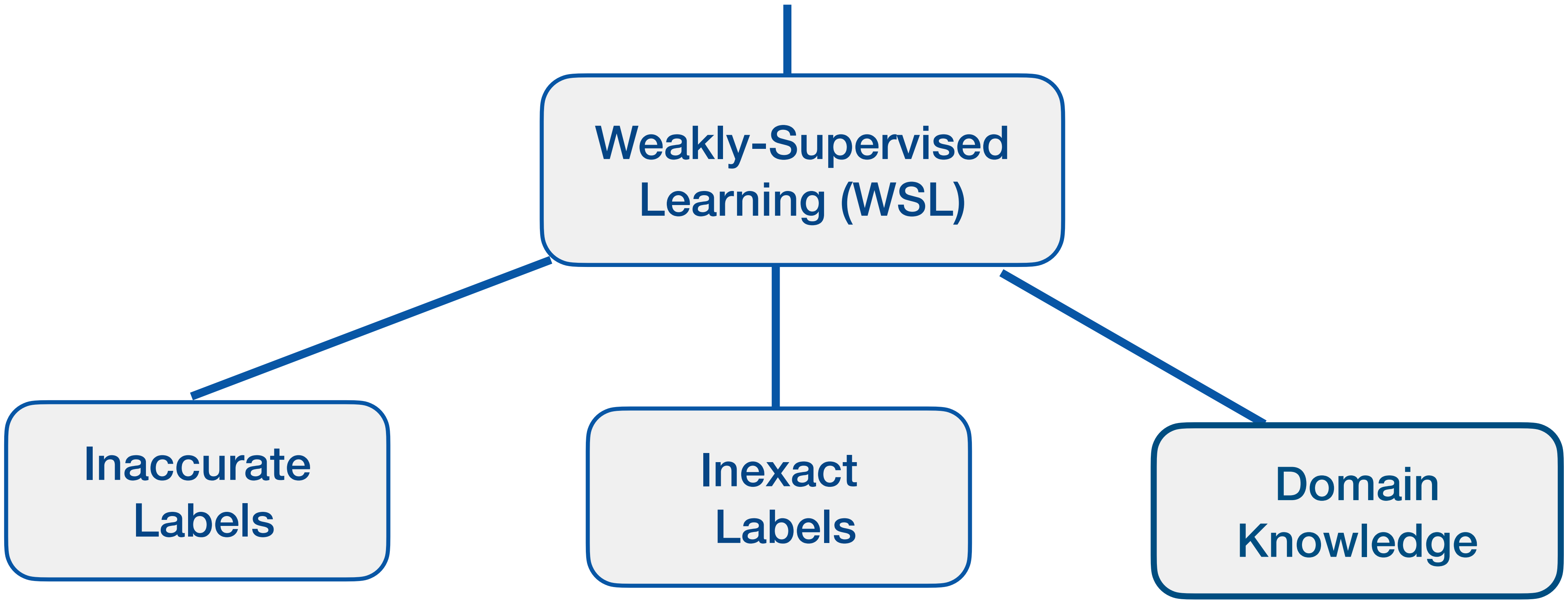


- Many successful applications of bootstrapping!
  - Seed **words** for information extraction [Riloff & Jones, 1999]
  - Seed **rules** for named entity recognition [Collins & Singer, 1999]
  - Seed **tuples** for relation extraction [Agichtein & Gravano, 2000]

- **Warning:** Model is unable to correct its own errors
  - Use domain rules to evaluate **quality** of model outputs
  - Then, discard "bad" model outputs

[Agichtein & Gravano, 2000]

WSL - Summary



- **WSL:**
  - Leverage inaccurate labels / inexact labels / prior domain knowledge …
  - … as weak supervision during learning

Taxonomy

Minimally Supervised Learning

Semi-Supervised Learning (SSL)
Leveraging unlabeled data

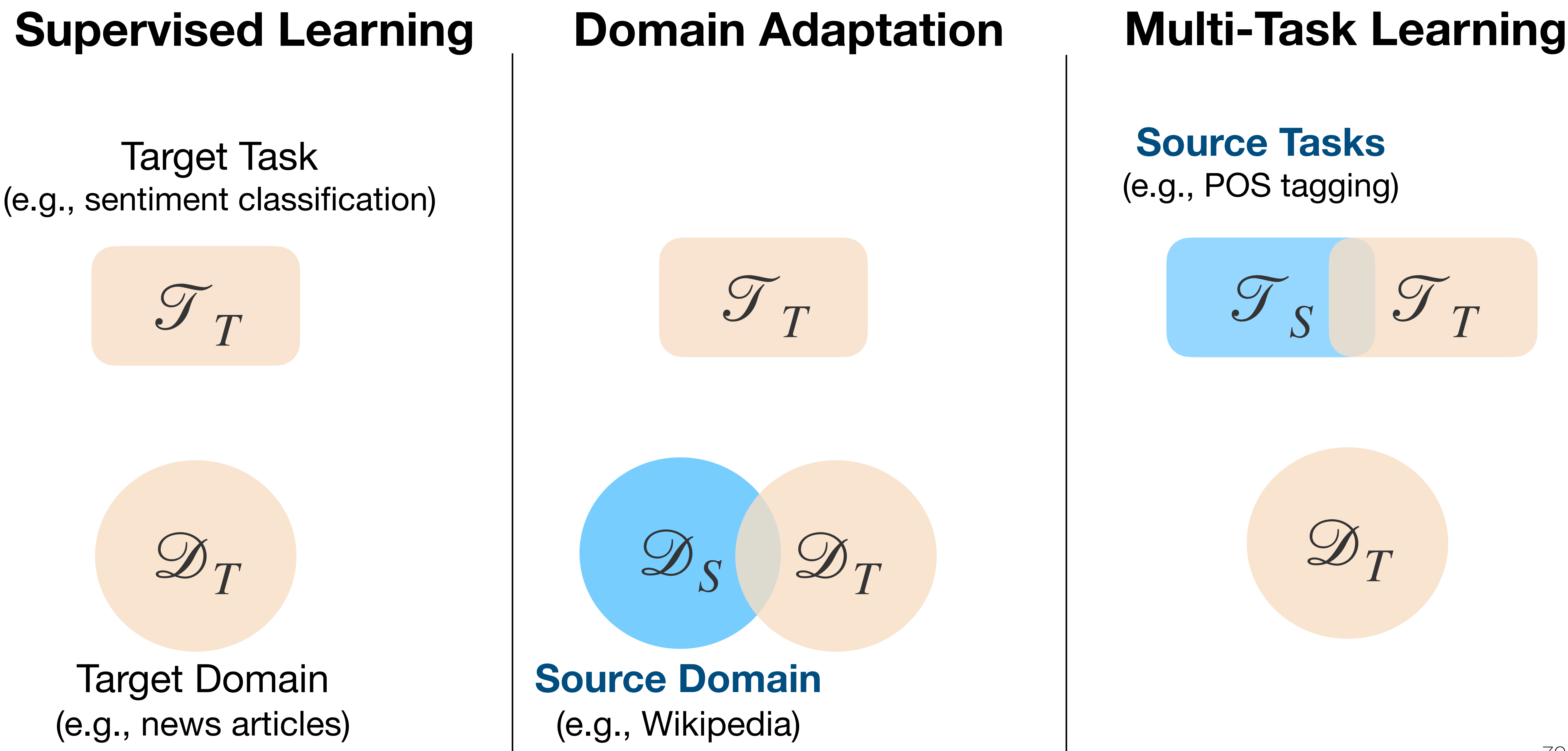
Weakly-Supervised Learning (WSL)
Leveraging weak labels / domain knowledge

Transfer Learning (TL)
Leveraging auxiliary domains / tasks

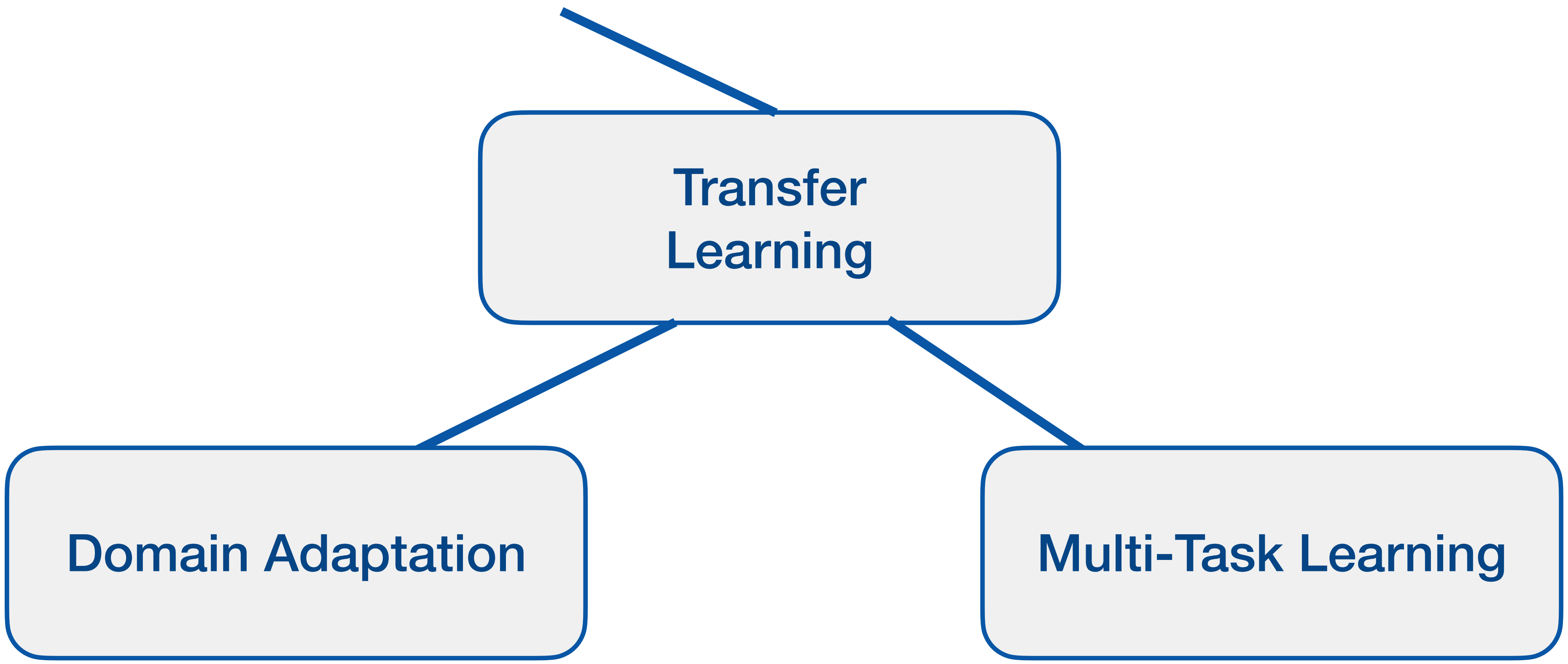
[Daumé, 2007]
[Collobert & Weston, 2008]
[Wan, 2009]
[Kim, 2014]
[Ammar et al., 2016]
[Peters et al., 2018]
[Howard & Ruder, 2018]
[Devlin et al., 2019]

Transfer Learning (TL)

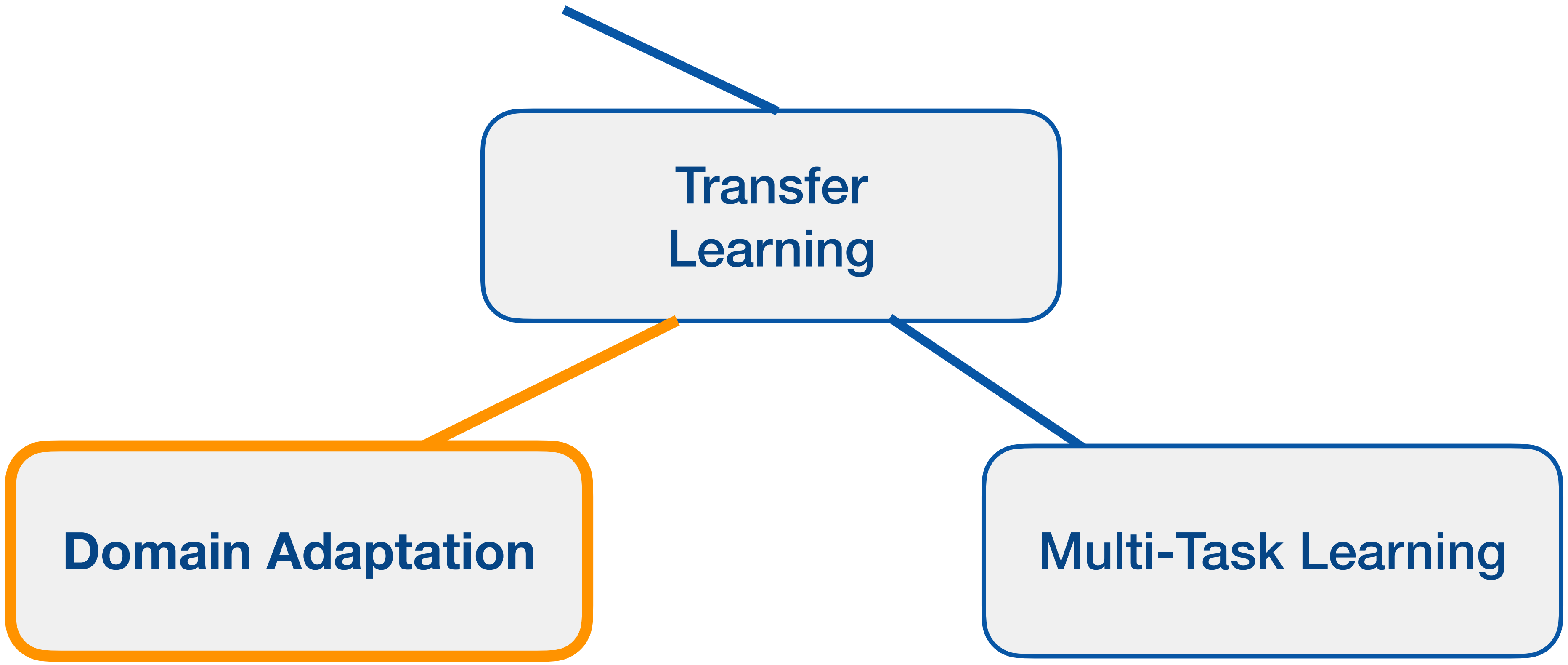
- **Transfer Learning:**
  - Leveraging auxiliary **domains** (domain adaptation)
  - Leveraging auxiliary **tasks** (multi-task learning)



Transfer Learning (TL) Taxonomy



Transfer Learning (TL) Taxonomy



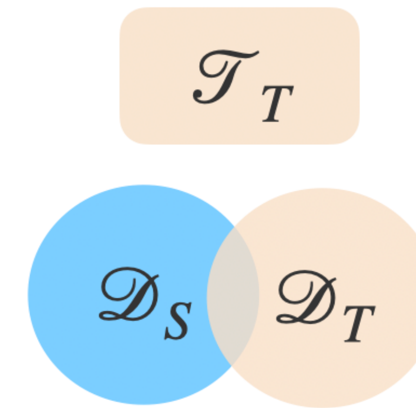
[Daumé, 2007]
[Wan, 2009]
[Ammar et al., 2016]

Domain Adaptation

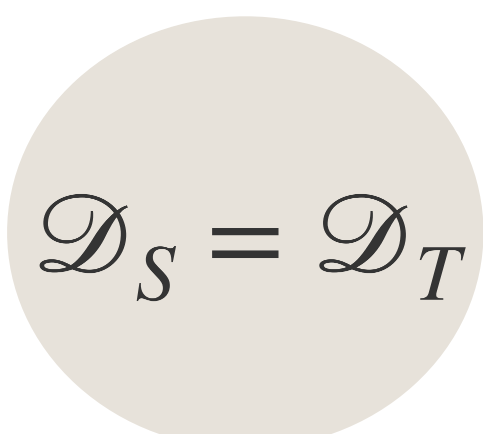
- **Goal**:
  - Improve performance in $\mathscr{D}_T$
  - … by “transferring knowledge” from $\mathscr{D}_S$

limited or no labeled data

many labeled data

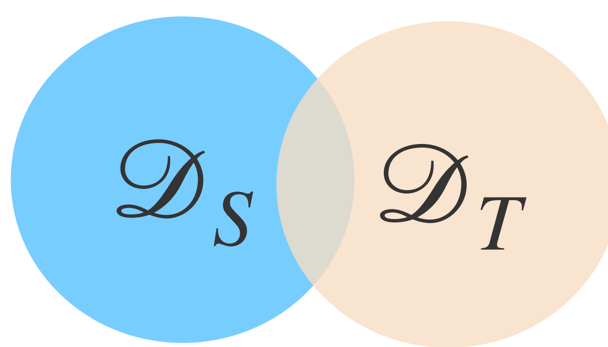


Ideal Scenario



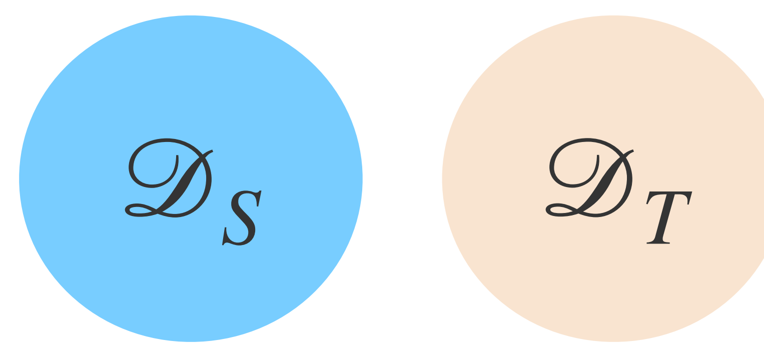
Supervised Learning

Real Scenario



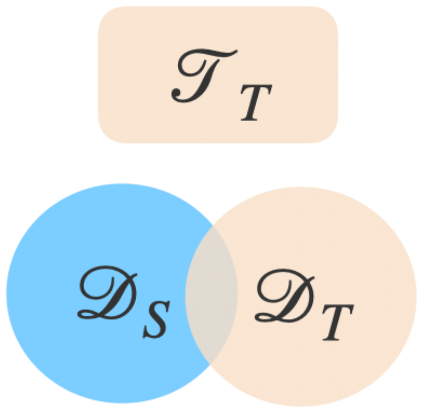
Predictive features in $\mathscr{D}_S$ could be useful in $\mathscr{D}_T$

Worst Scenario



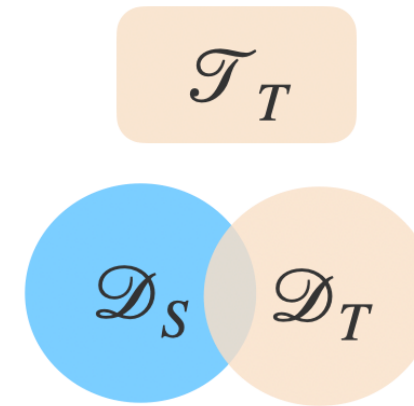
$\mathscr{D}_S$ is not useful

How to Leverage Source Domain?



- Domain: $\mathcal{D}(\mathcal{X}, P(X))$
    - $\mathcal{X}$ = feature space (e.g., English n-grams)
    - $P(X)$ = marginal probability distribution

How to Leverage Source Domain?



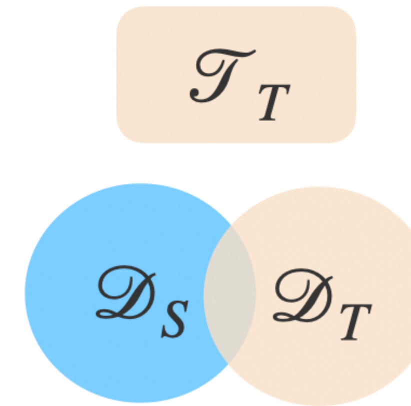
- Domain: $\mathcal{D}(\mathcal{X}, P(X))$
  - $\mathcal{X}$ = feature space (e.g., English n-grams)
  - $P(X)$ = marginal probability distribution

- Even simple "feature augmentation" approach is effective [Daumé III, 2007]
  - Create **multiple copies** of each feature: "shared" and "domain-specific"

$$\begin{aligned} \mathcal{X}_S &= \mathcal{X}_{SHARED} + \mathcal{X}_{S-SPECIFIC} \\ \mathcal{X}_T &= \mathcal{X}_{SHARED} + \mathcal{X}_{T-SPECIFIC} \end{aligned} \quad \Rightarrow \quad \mathcal{X}_S \cup \mathcal{X}_T = \mathcal{X}_{SHARED} + \mathcal{X}_{S-SPECIFIC} + \mathcal{X}_{T-SPECIFIC}$$

  - Model trained $\mathcal{D}_S \cup \mathcal{D}_T$ and encouraged to rely on $\mathcal{X}_{SHARED}$
  - **Effect:** $\mathcal{X}_{SHARED}$ after training are **better parameter estimates** for $\mathcal{T}_T$

How to Leverage Source Domain?



- Domain: $\mathcal{D}(\mathcal{X}, P(X))$
  - $\mathcal{X}$ = feature space (e.g., English n-grams)
  - $P(X)$ = marginal probability distribution

- Even simple "feature augmentation" approach is effective [Daumé III, 2007]
  - Create **multiple copies** of each feature: "shared" and "domain-specific"

$$\mathcal{X}_S = \mathcal{X}_{SHARED} + \mathcal{X}_{S-SPECIFIC}$$
$$\mathcal{X}_T = \mathcal{X}_{SHARED} + \mathcal{X}_{T-SPECIFIC}$$
$$\rightarrow \mathcal{X}_S \cup \mathcal{X}_T = \mathcal{X}_{SHARED} + \mathcal{X}_{S-SPECIFIC} + \mathcal{X}_{T-SPECIFIC}$$

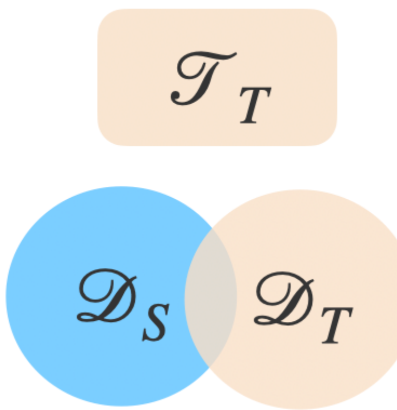
  - Model trained $\mathcal{D}_S \cup \mathcal{D}_T$ and encouraged to rely on $\mathcal{X}_{SHARED}$
  - **Effect:** $\mathcal{X}_{SHARED}$ after training are **better parameter estimates** for $\mathcal{T}_T$

(-) expensive: requires **many** target labels (labels in $\mathcal{D}_T$)

How to Leverage Source Domain?

- Further approaches rely on **fewer or no** target labels
  - **Main idea:** bring representations from D_S , D_T closer
  - **Objective:** min_dist($\mathscr{D}_S$, $\mathscr{D}_T$) + max_performance($\mathscr{D}_S$)

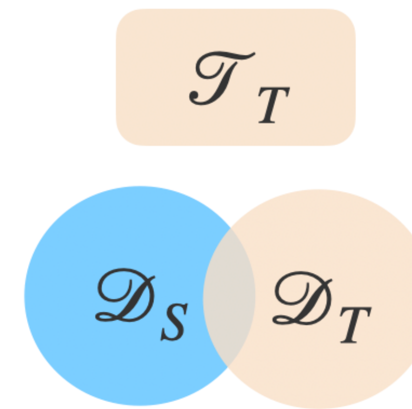
only unlabeled data **source labeled data**



How to Leverage Source Domain?

- Further approaches rely on **fewer or no** target labels
  - **Main idea:** bring representations from D_S , D_T closer
  - **Objective:** min_dist($\mathcal{D}_S$, $\mathcal{D}_T$) + max_performance($\mathcal{D}_S$)

    only unlabeled data **source labeled data**
  - More "distant" domains -> harder problem [Blitzer, 2007]



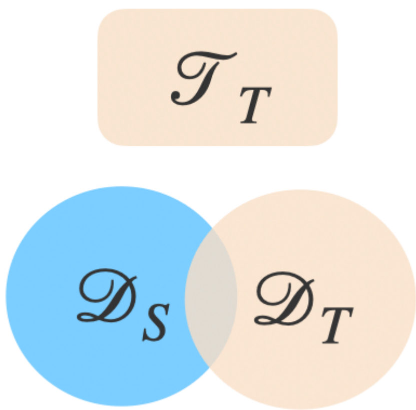
How to Leverage Source Domain?

- Further approaches rely on **fewer or no** target labels
  - **Main idea:** bring representations from D_S , D_T closer
  - **Objective:** min_dist($\mathscr{D}_S$, $\mathscr{D}_T$) + max_performance($\mathscr{D}_S$)

    only unlabeled data **source labeled data**
  - More “distant” domains -> harder problem [Blitzer, 2007]

    (-) implicit assumption: overlap in feature spaces $\mathscr{X}_S \cap \mathscr{X}_T \supsetneq \emptyset$

    not always true!



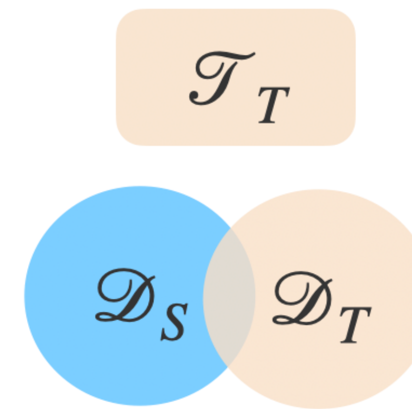
How to Leverage Source Domain?

- Further approaches rely on **fewer or no** target labels
  - **Main idea:** bring representations from D_S , D_T closer
  - **Objective:** min_dist($\mathscr{D}_S$, $\mathscr{D}_T$) + max_performance($\mathscr{D}_S$)

    only unlabeled data **source labeled data**
  - More "distant" domains -> harder problem [Blitzer, 2007]

    (-) implicit assumption: overlap in feature spaces $\mathscr{X}_S \cap \mathscr{X}_T \supsetneq \emptyset$

    not always true!



- **Cross-lingual learning**
  - Challenging: $\mathscr{X}_S \cap \mathscr{X}_T = \emptyset$ (or so)
  - How to align $\mathscr{X}_S, \mathscr{X}_T$?

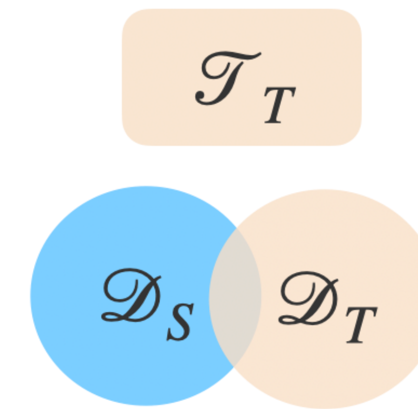
How to Leverage Source Domain?

- Further approaches rely on **fewer or no** target labels
  - **Main idea:** bring representations from D_S , D_T closer
  - **Objective:** min_dist($\mathcal{D}_S$, $\mathcal{D}_T$) + max_performance($\mathcal{D}_S$)

    only unlabeled data **source labeled data**
  - More "distant" domains -> harder problem [Blitzer, 2007]

    (-) implicit assumption: overlap in feature spaces $\mathcal{X}_S \cap \mathcal{X}_T \supsetneq \emptyset$

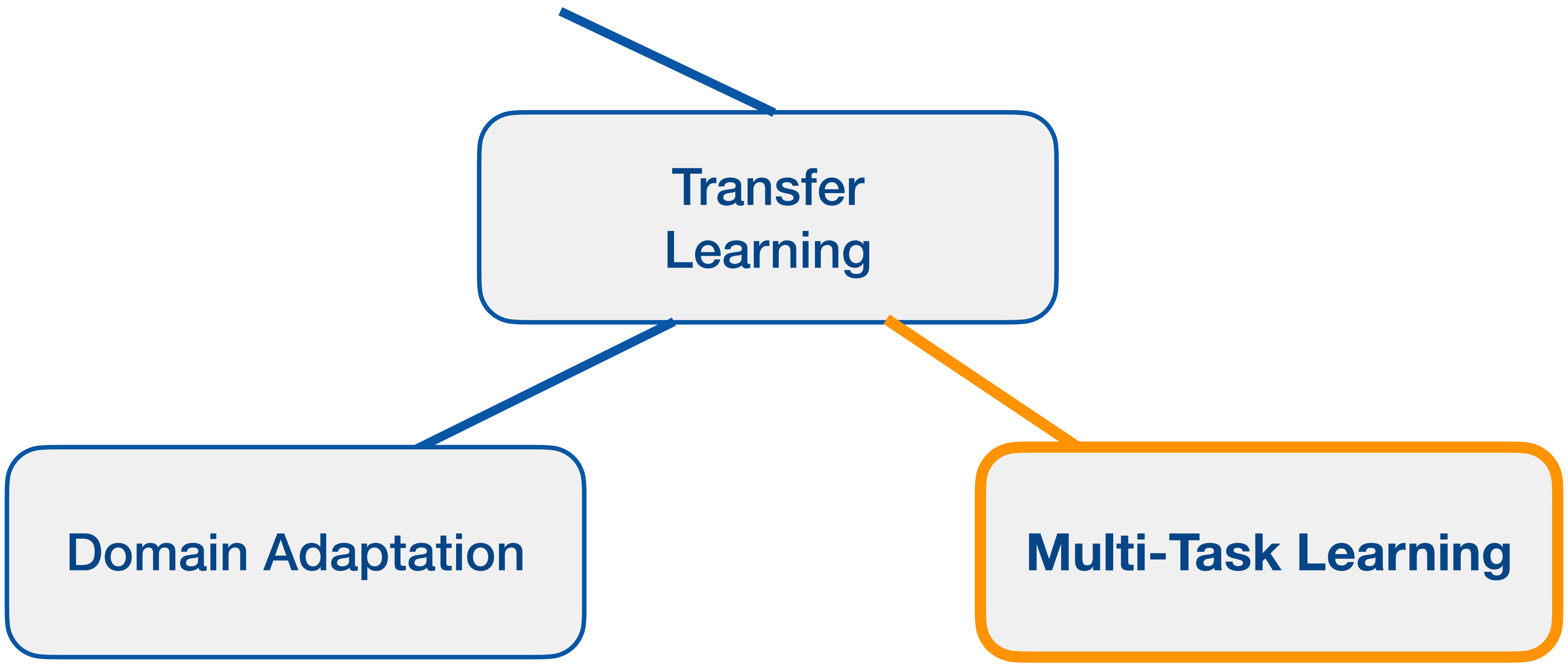
    not always true!



- **Cross-lingual learning**
  - Challenging: $\mathcal{X}_S \cap \mathcal{X}_T = \emptyset$ (or so)
  - How to align $\mathcal{X}_S, \mathcal{X}_T$?

Resource:	Parallel Documents →	Machine Translation →	Bilingual Dictionaries →	Seed Dictionaries
Cons:	**not always available**	**expensive & error-prone**	**not always available**	**effective?**
		[Wan, 2009]	[Ammar et al., 2016]	[Artetxe et al., 2018]

Transfer Learning (TL) Taxonomy

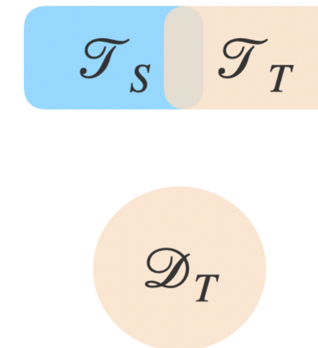


Multi-Task Learning (MTL)

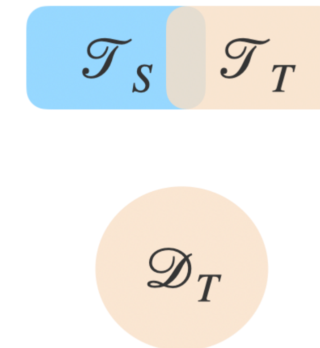
- **Goal**:
  - Improve performance for $\mathscr{T}_T$
  - … by leveraging training data from source tasks $\mathscr{T}_S$

limited or no labeled data

many labeled data



Multi-Task Learning (MTL)

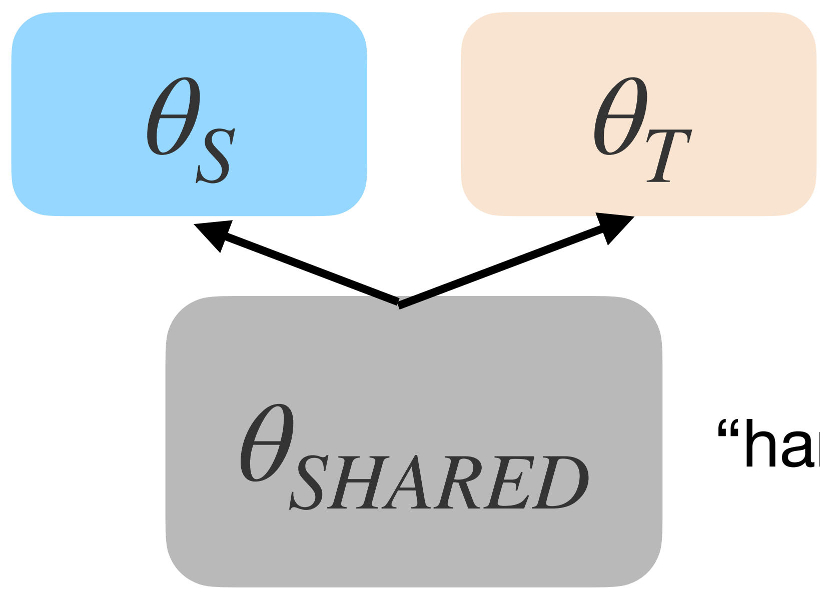


- **Goal**:
  - Improve performance for $\mathcal{T}_T$

    limited or no labeled data
  - … by leveraging training data from source tasks $\mathcal{T}_S$

    many labeled data
- **Common practice:** share representation across tasks

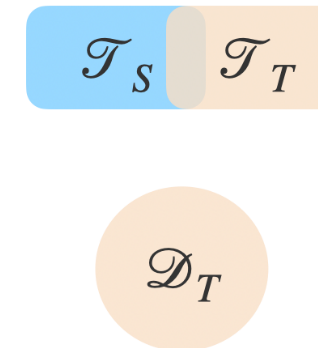
Model Architecture



[Collobert & Weston, 2008]

"hard sharing"

Multi-Task Learning (MTL)

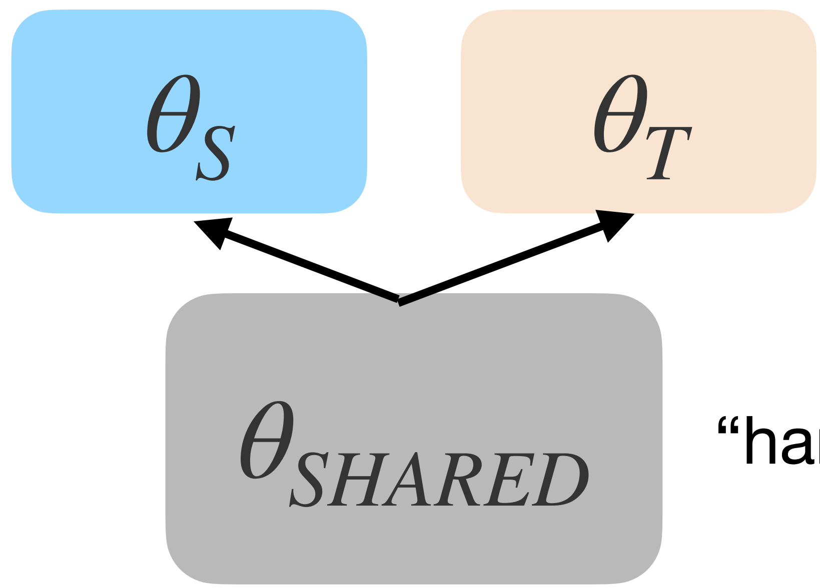


- **Goal**:
  - Improve performance for $\mathscr{T}_T$ ← **limited or no labeled data**
  - … by leveraging training data from source tasks $\mathscr{T}_S$ ← **many labeled data**

- **Common practice:** share representation across tasks

[Collobert & Weston, 2008]

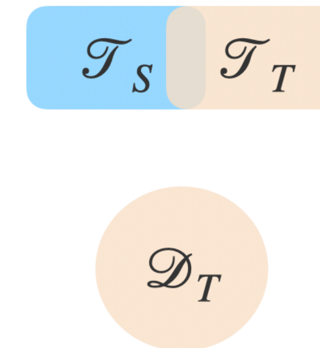
Model Architecture



"hard sharing"

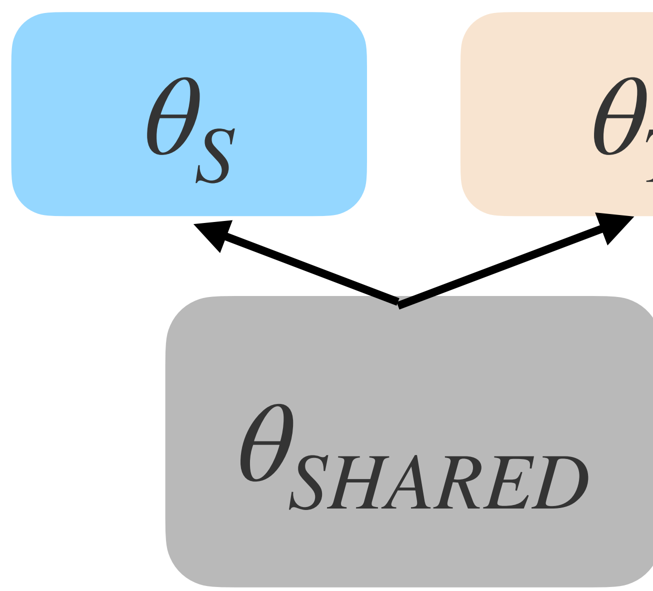
- **Why does MTL work?**
  - Training signals in $\mathscr{T}_S$ could improve generalization in $\mathscr{T}_T$ [Caruana et al., 1997]
  - θ_{SHARED}: effectively see more data

Multi-Task Learning (MTL)



- **Goal**:
  - Improve performance for $\mathscr{T}_T$ ← **limited or no labeled data**
  - … by leveraging training data from source tasks $\mathscr{T}_S$ ← **many labeled data**
- **Common practice:** share representation across tasks

Model Architecture



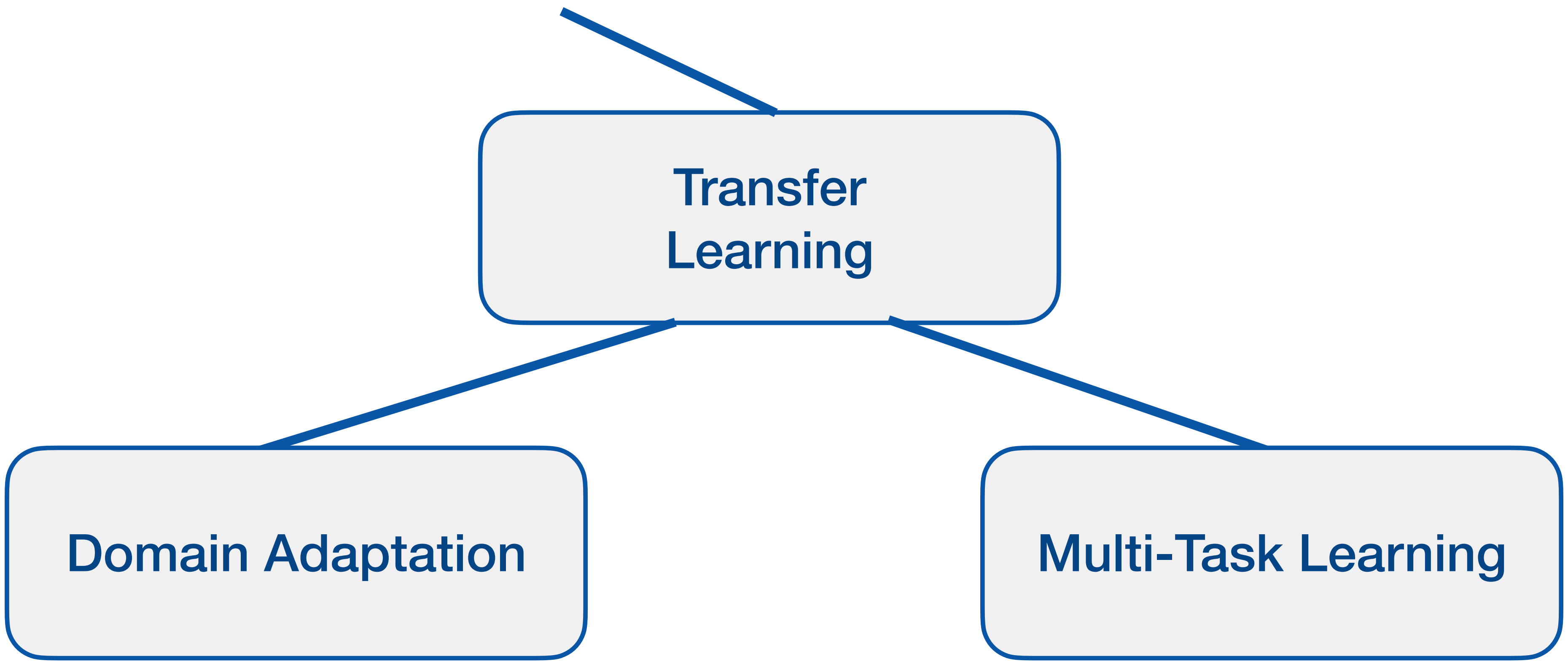
[Collobert & Weston, 2008]

"hard sharing"

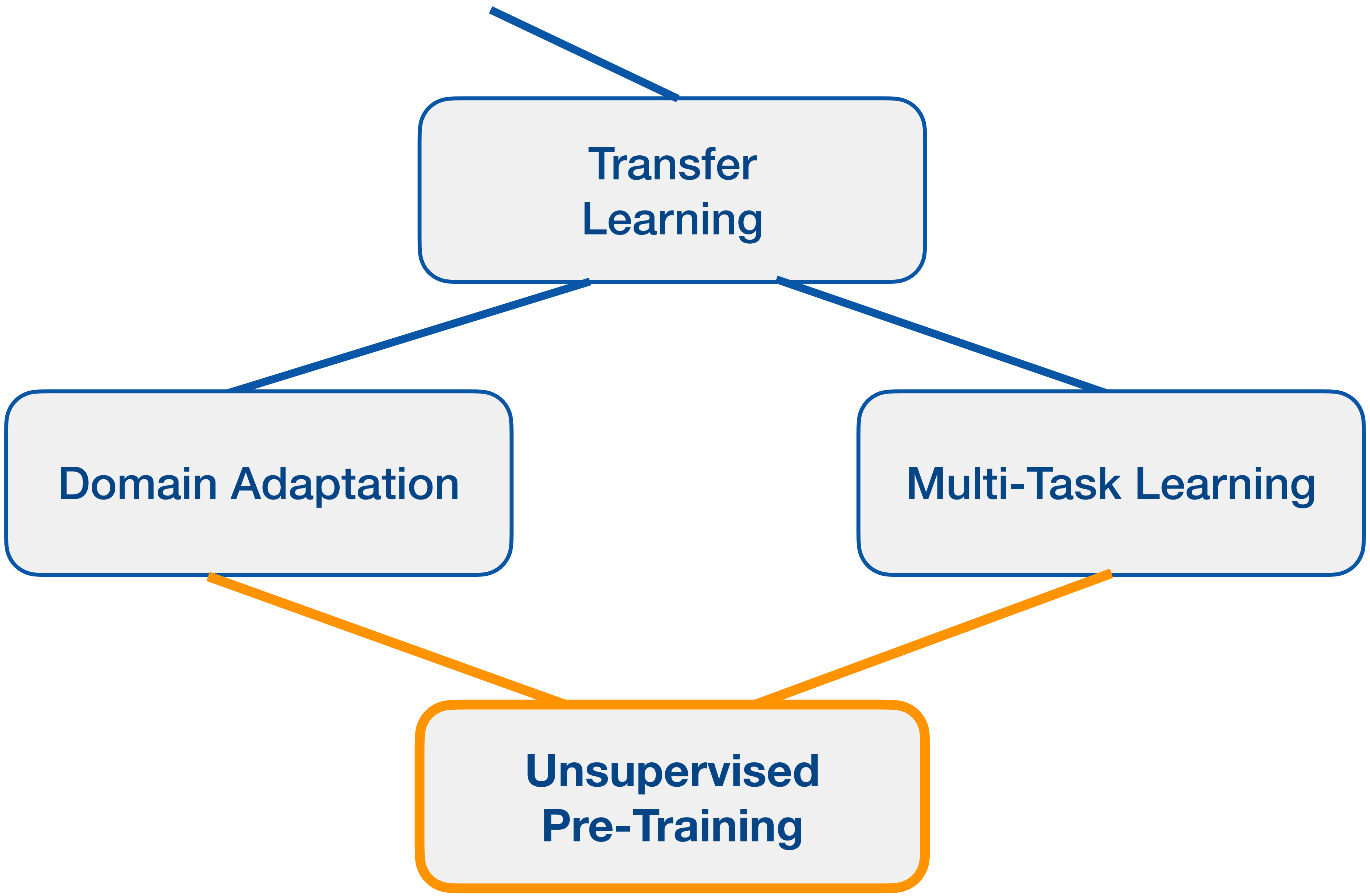
- **Why does MTL work?**
  - Training signals in $\mathscr{T}_S$ could improve generalization in $\mathscr{T}_T$ [Caruana et al., 1997]
  - θ_{SHARED}: effectively see more data

(-) Caveat: inefficient for big tasks as all source data required for target training

Transfer Learning (TL) Taxonomy



Transfer Learning (TL) Taxonomy



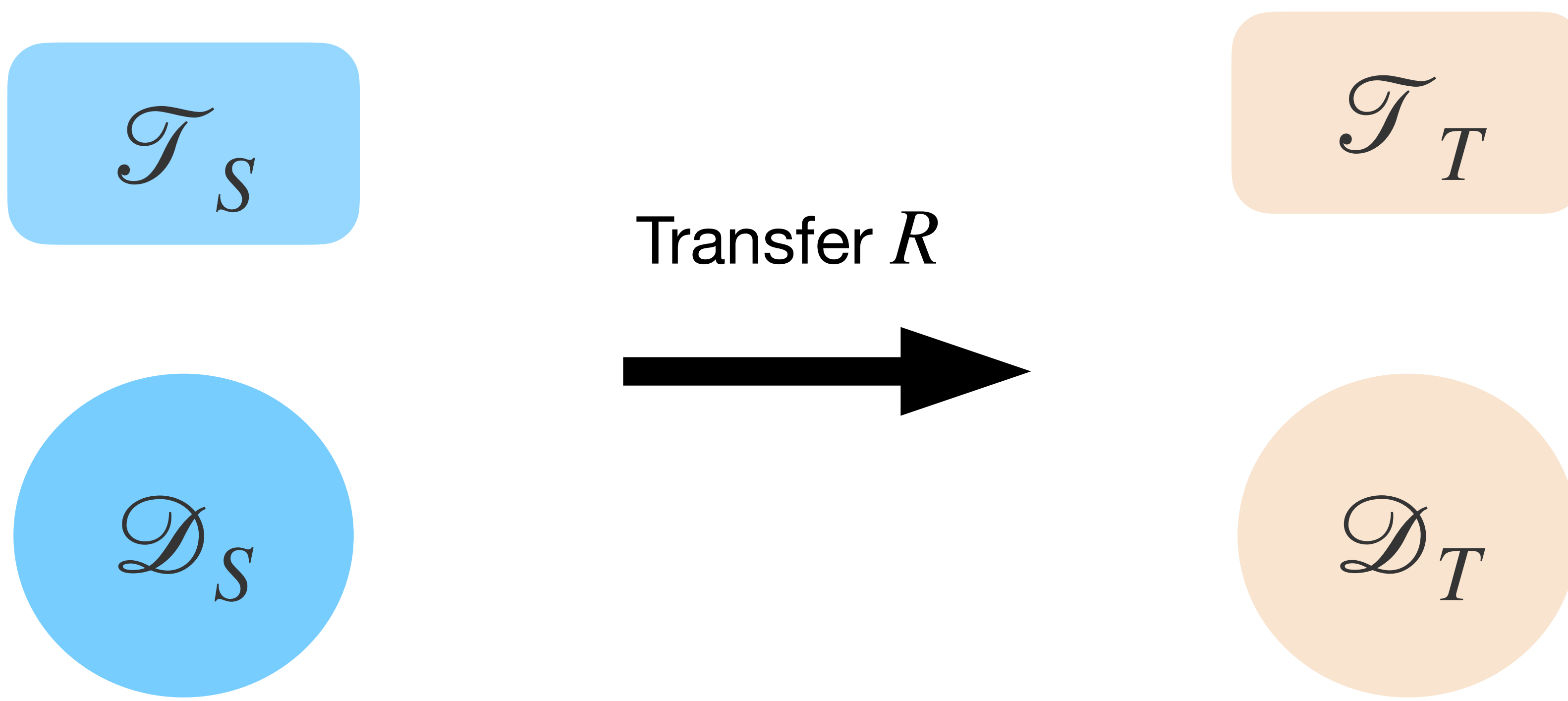
[Collobert & Weston, 2008]
[Kim, 2014]
[Ammar et al., 2016]
[Peters et al., 2018]
[Howard & Ruder, 2018]
[Devlin et al., 2019]

Unsupervised Pre-Training

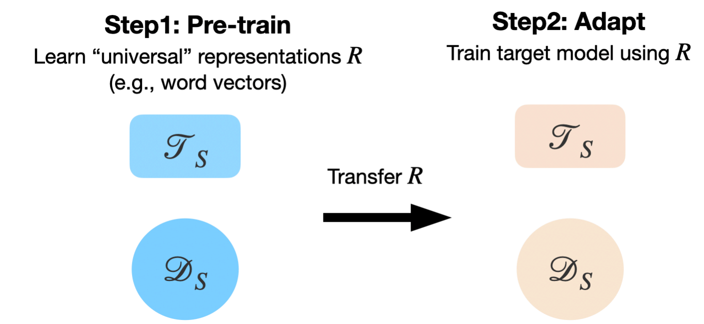
- Sequential transfer learning approach:

Step1: Pre-train
Learn “universal” representations R

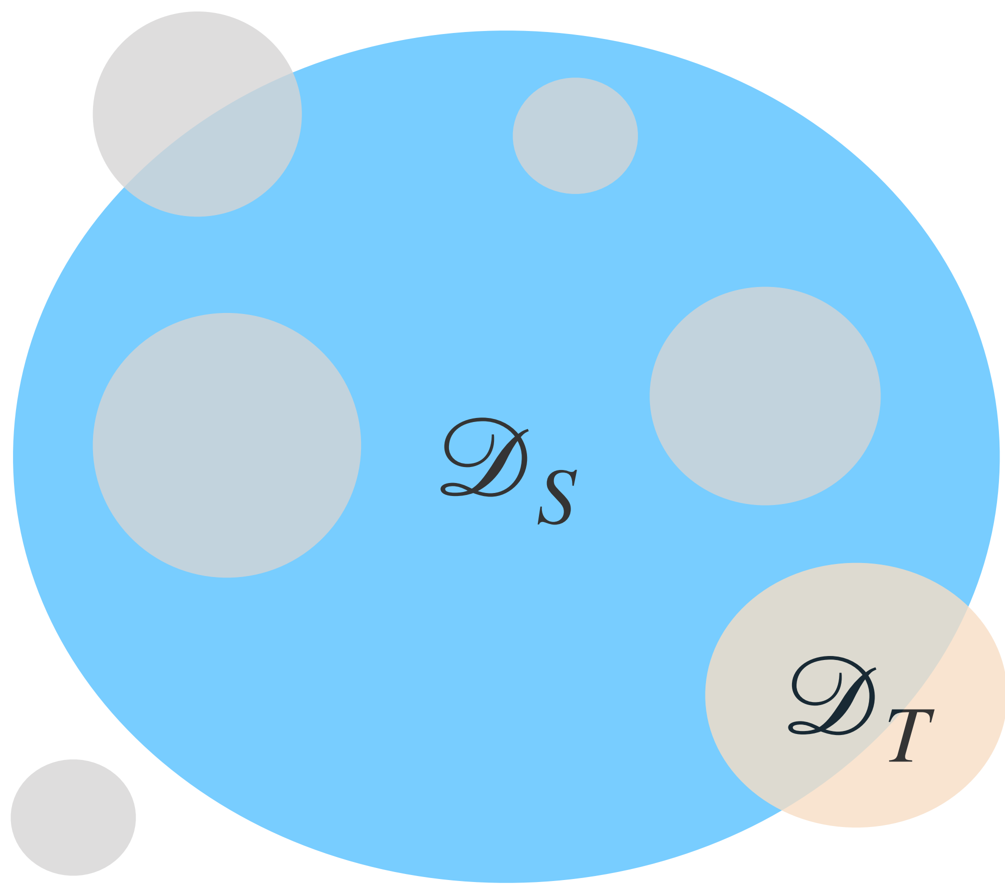
Step2: Adapt
Train target model using R



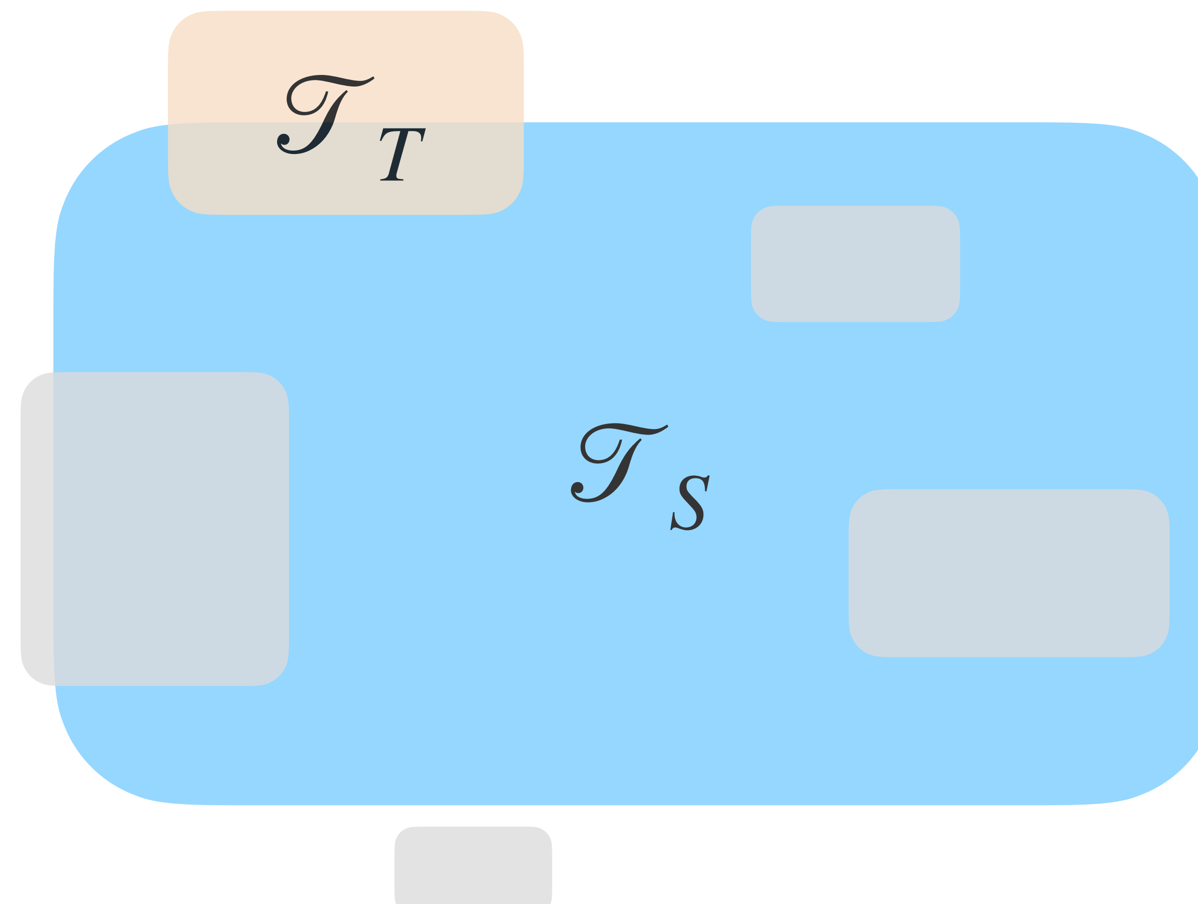
Explaining the Effectiveness of Unsupervised Pre-Training



- **Why** should unsupervised pre-training work?
  - Because of **"universal"** R
  - R captures general aspects of language structure/meaning
  - R = useful features for θ_T : no need to re-learn from scratch



$\mathscr{D}_{\mathcal{S}}$: very big domain
(e.g., Wikipedia)



$\mathscr{T}_{\mathcal{S}}$: unsupervised objective
(e.g., language modeling)

Common Practices in Unsupervised Pre-Training Step
From Static to Contextual Representations

[Mikolov et al. 2013]
[Pennington et al. 2014]

- **Early approaches:** learn "static" word vectors R

  (-) limited expressiveness:
  - R may **not encode** compositional meaning (e.g., negation)
  - θ_T may need more data to re-learn word composition **from scratch**

Common Practices in Unsupervised Pre-Training Step
From Static to Contextual Representations

- **Early approaches:** learn "static" word vectors R

[Mikolov et al. 2013]
[Pennington et al. 2014]

(-) limited expressiveness:

- R may **not encode** compositional meaning (e.g., negation)
- θ_T may need more data to re-learn word composition **from scratch**

- **Recent approaches:** learn "contextual" language representations R

1. Pre-train **deep** language model
2. Transfer **all layers**

[Peters et al. 2018]
[Howard & Ruder, 2018]
[Devlin et al. 2019]

(+) Capture more complex language phenomena [Peters et al. 2018]

- Lower layers may capture syntax
- Upper layers may capture long-range dependencies (e.g., coreference)

(-) Computationally expensive: many GPU days & billions of parameters

(+) BUT: you (?) only pre-train once!

Common Practices in Adaptation Step
Feature Extraction Vs Fine-Tuning

[Kim, 2014]
[Peters et al., 2018]
[Devlin et al. 2019]

- **“Feature extraction”:** use R as “frozen” features
  - **(+) computational efficiency:** save space & time
  - **(-) limited effectiveness:**
    - task-specific features may not be captured (e.g., for rare events)

Common Practices in Adaptation Step
Feature Extraction Vs Fine-Tuning

[Kim, 2014]
[Peters et al., 2018]
[Devlin et al. 2019]

- **"Feature extraction":** use R as "frozen" features

  (+) computational efficiency: save space & time

  (-) limited effectiveness:

  - task-specific features may not be captured (e.g., for rare events)

[Kim, 2014]
[Howard & Ruder, 2018]
[Devlin et al. 2019]

- **"Fine-tuning":** update R during training θ_T

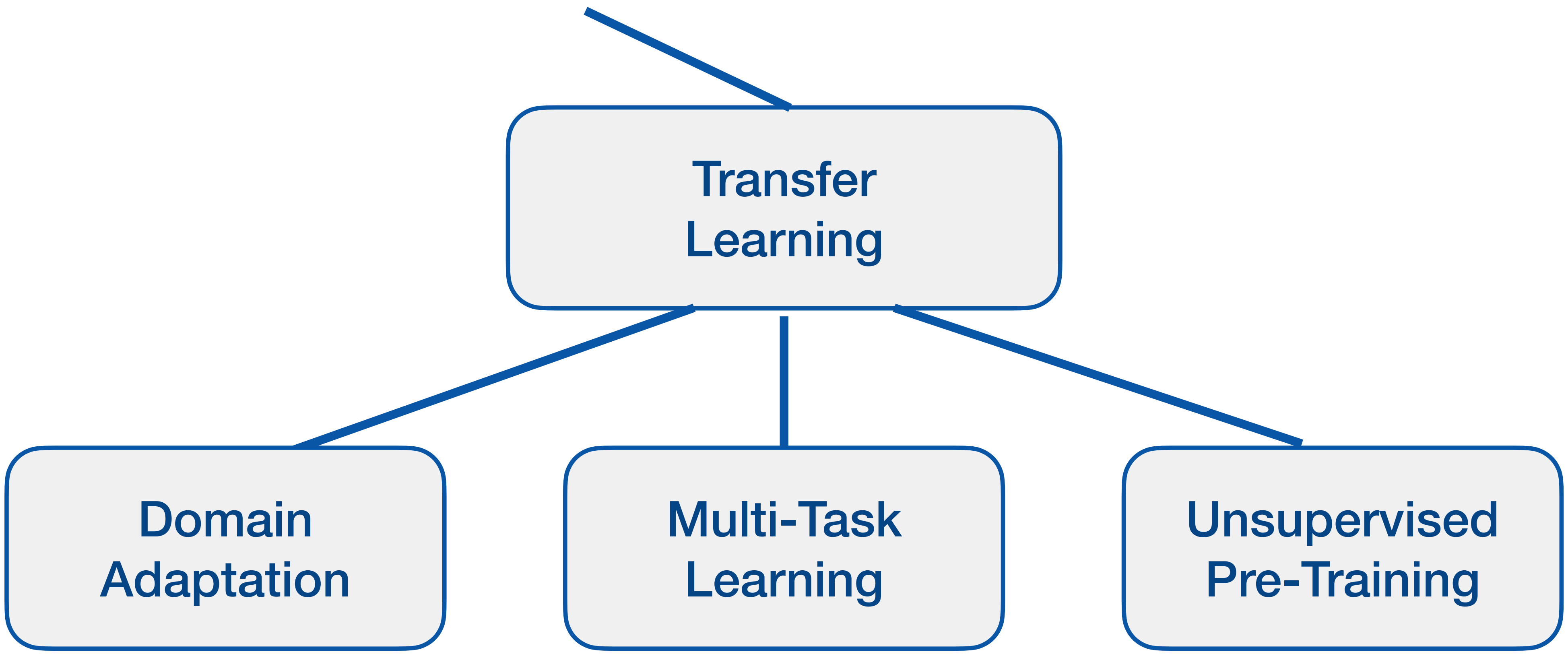
  (+) effectiveness: general -> task-specific representations

  (-) expensive

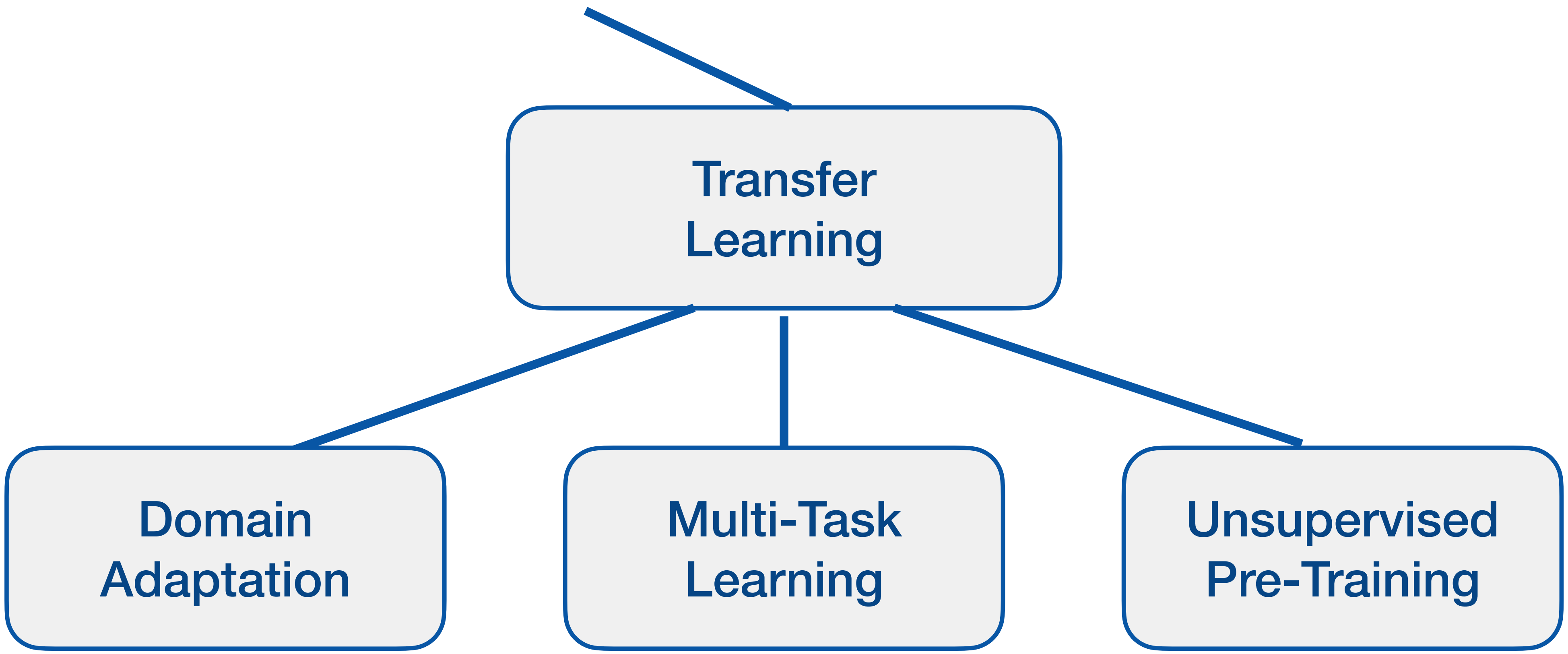
  (-) risk of overfitting in limited labeled data settings [Howard & Ruder, 2018]

  - "Lack of knowledge of how to train [language models] effectivey"
  - Fine-tuning tricks: "gradual unfreezing", "slanted triangular learning rates",…

Transfer Learning Summary



Transfer Learning Summary



- **Caveat:** Transfer learning could **hurt** performance (negative transfer)
  - Most approaches **implicitly** assume **related** task/domains [Pan & Yang, 2009]
  - Answer "**what**" & "**how**" to transfer. Not "**when**"

Taxonomy

Minimally Supervised Learning

Semi-Supervised Learning (SSL)

Leveraging unlabeled data

Generative

- EM-based

Discriminative

- Clustering-based
- Co-training-based

Weakly-Supervised Learning (WSL)

Leveraging weak labels / domain knowledge

Inaccurate Labels

- Crowdsourcing
- Learning with Noisy Labels

Inexact Labels

- Multiple Instance Learning

Domain Knowledge

- Posterior Regularization
- Data Programming
- Bootstrapping

Transfer Learning (TL)

Leveraging auxiliary domains / tasks

Domain Adaptation

- Feature Alignment
- Cross-Lingual Learning

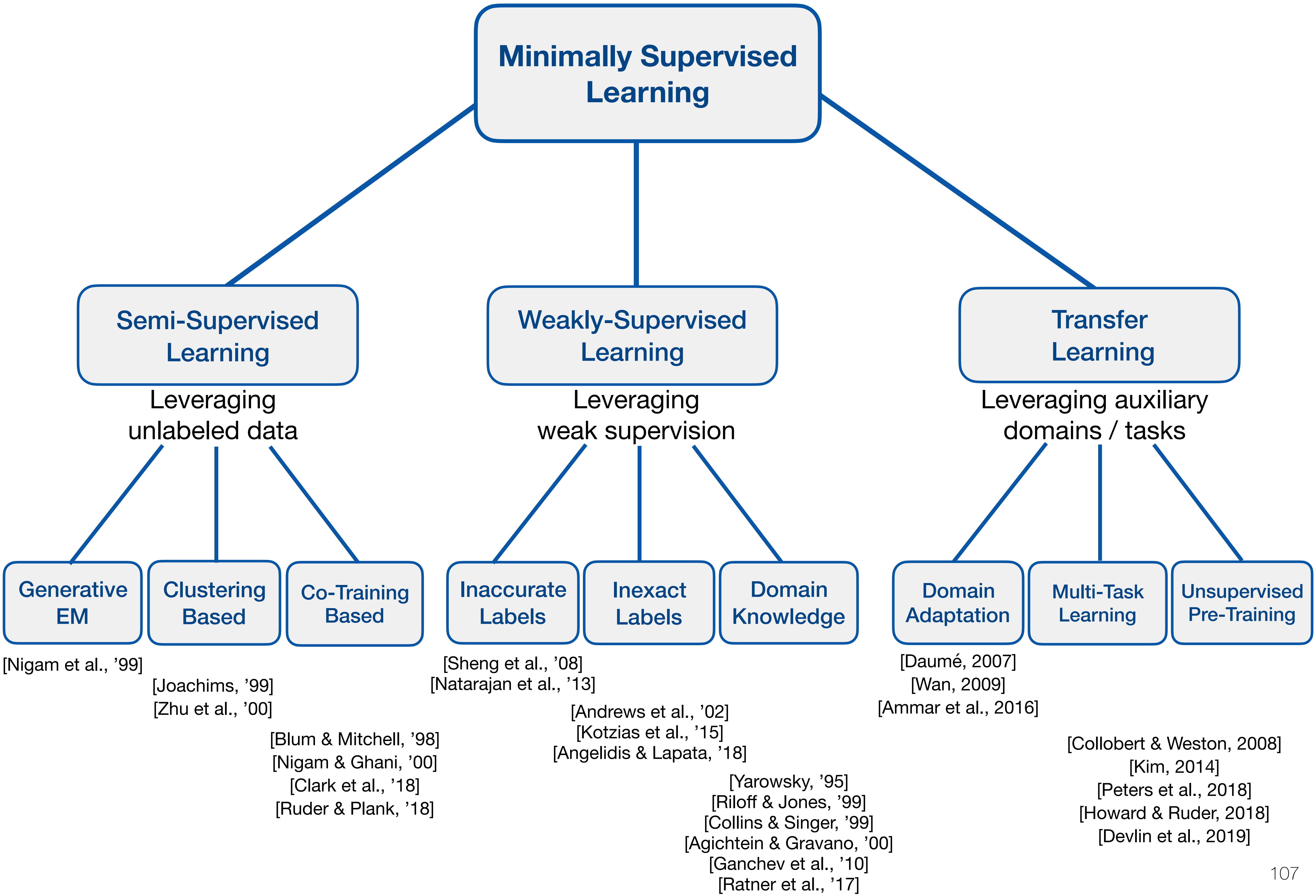
Multi-Task Learning

- Weight Sharing

Unsupervised Pre-Training

- Pre-training
- Adaptation

Full Taxonomy & Papers



Thank you!

gkaraman@cs.columbia.edu
https://gkaramanolakis.github.io

COLUMBIA UNIVERSITY