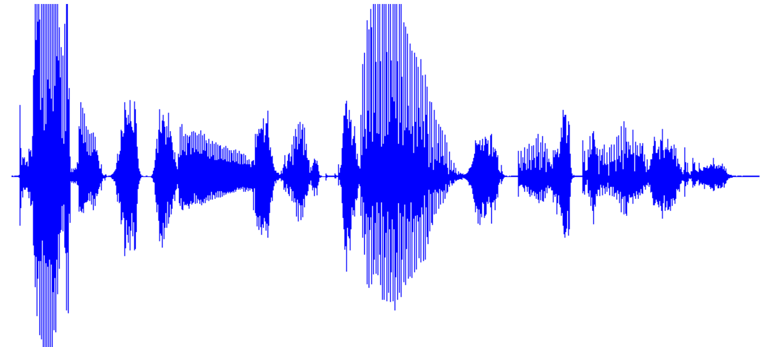
Domain Adaptation with Structural Correspondence Learning

John Blitzer

Joint work with

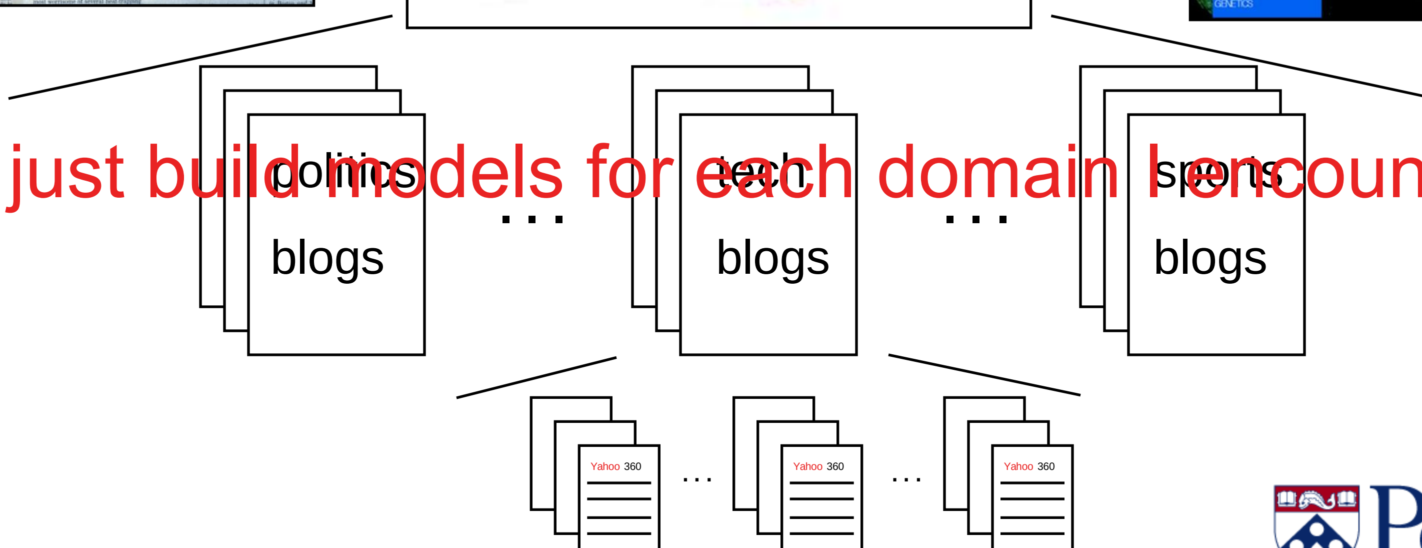
Shai Ben-David, Koby Crammer, Mark Dredze, Ryan McDonald, Fernando Pereira

Statistical models, multiple domains



Different Domains of Text

- Huge variation in vocabulary & style

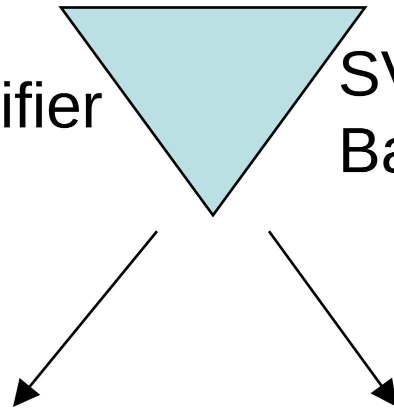


"Ok, I'll just build models for each domain I encounter"

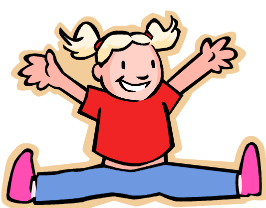
Sentiment Classification for Product Reviews

Product Review

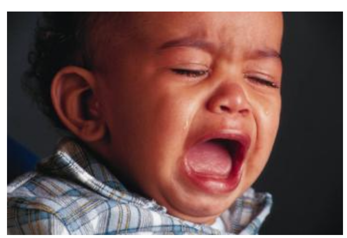
Classifier



SVM, Naïve Bayes, etc.



Positive

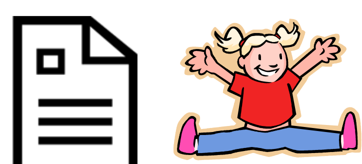


Negative

Multiple Domains

books

. . .



kitchen appliances

??

??

??

books & kitchen appliances

Running with Scissors: A Memoir

Title: Horrible book, horrible.

This book was horrible. I **read half** of it, **suffering from a headache** the entire time,

copy in the world....don't waste your money. I wish i had the time spent reading this book back so i could use it for better purposes. This book wasted my life

Avante Deep Fryer, Chrome & Black

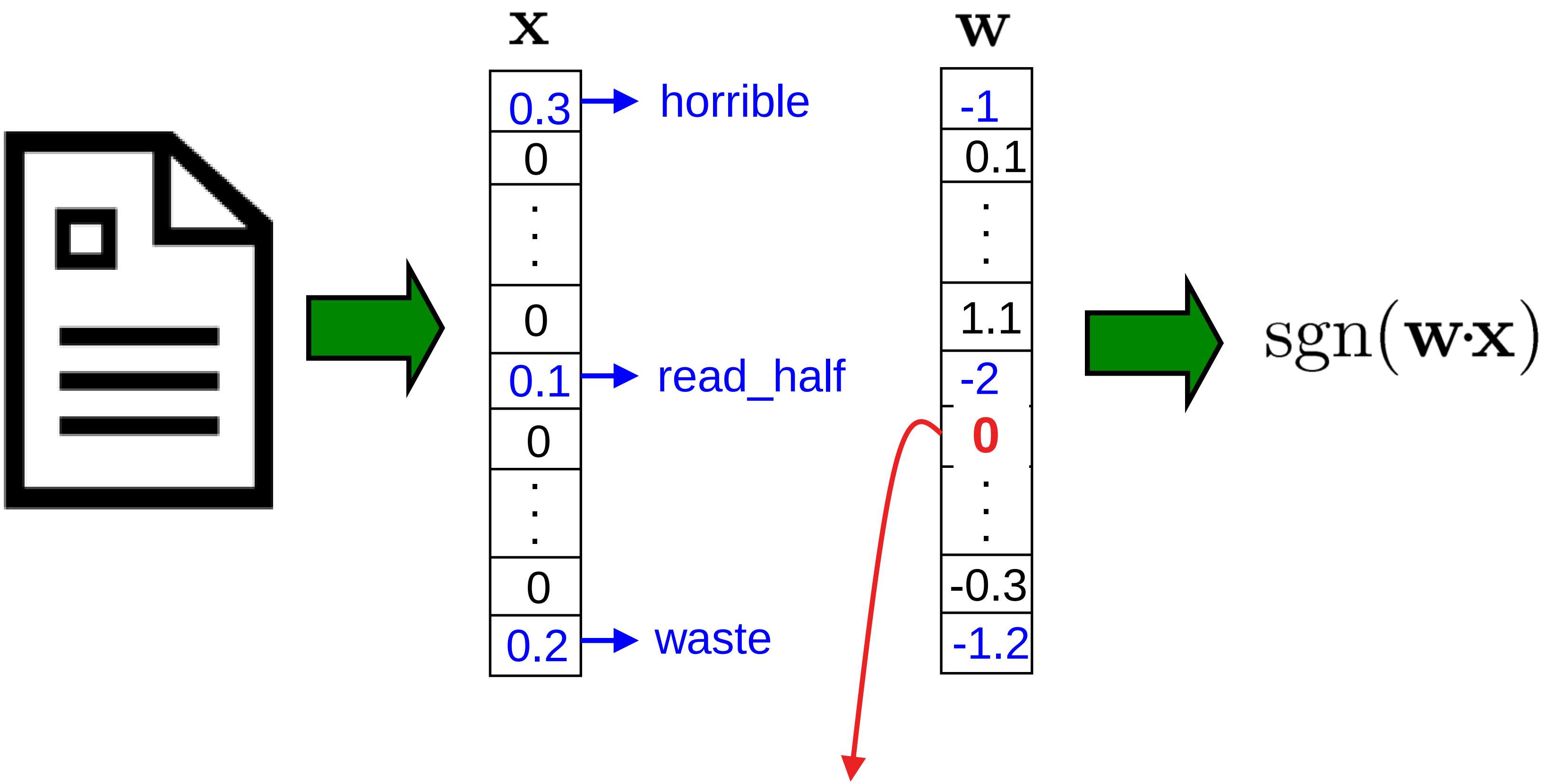
Title: lid does not work well...

I love the way the Tefal deep fryer cooks, however, I am **returning** my

closure. The lid may close initially, but after a few uses it no longer stays closed. I **will not be purchasing** this one again.

Error increase: 13% → 26%

Features & Linear Models



Problem: If we've only trained on book reviews, then **w(defective) = 0**

Structural Correspondence Learning (SCL)

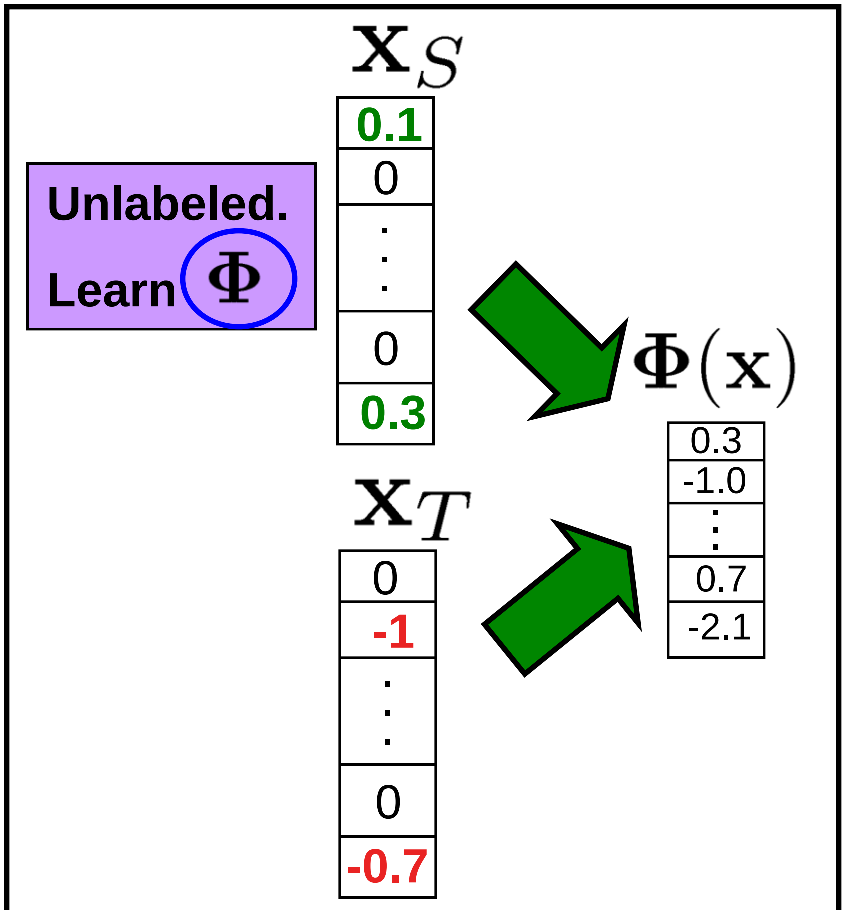
- **Cut adaptation error by more than 40%**
- Use **unlabeled** data from the target domain
- Induce correspondences among different features
- **read-half, headache** $\longleftrightarrow$ **defective, returned**
- Labeled data for **source** domain will help us build a good classifier for **target** domain

Maximum likelihood linear regression (MLLR) for speaker adaptation (Leggetter & Woodland, 1995)

Penn
University of Pennsylvania

SCL: 2-Step Learning Process

Step 1: Unlabeled – Learn correspondence mapping



Step 2: Labeled – Learn weight vector

Labeled. Learn $\mathbf{v}$

$$\Phi(\mathbf{x}) \Rightarrow \mathrm{sgn}\left(\mathbf{v}\cdot\Phi(\mathbf{x})\right)$$

- **Φ should make the domains look as similar as possible**
- **But Φ should also allow us to classify well**

SCL: Making Domains Look Similar

Incorrect classification of kitchen review **defective** lid

Unlabeled **kitchen** contexts

- Do **not buy** the Shark portable steamer …. Trigger mechanism is **defective**.
- the very nice lady assured me that I must have a **defective** set …. What a **disappointment**!
- Maybe mine was **defective** …. The directions were **unclear**

Unlabeled **books** contexts

- The book is so **repetitive** that I found myself yelling …. I will definitely **not buy** another.
- A **disappointment** …. Ender was talked about for **<#> pages** altogether.
- it’s **unclear** …. It’s repetitive and **boring**

SCL: Pivot Features

Pivot Features

- Occur frequently in both domains
- Characterize the task we want to do
- Number in the hundreds or thousands
- Choose using labeled **source**, unlabeled **source** & **target** data

SCL: words & bigrams that occur frequently in both domains

SCL-MI: SCL but also based on mutual information with labels

book one <num> so all very about they like good when	**a_must a_wonderful loved_it weak don't_waste awful highly_recommended and_easy**

SCL Unlabeled Step: Pivot Predictors

Use **pivot features** to align other features

(1) The book is so **repetitive** that I found myself yelling …. I will definitely another.	**(2)** Do the Shark portable steamer …. Trigger mechanism is **defective**.

Binary problem: Does "**not buy**" appear here?

- **Mask** and predict pivot features using other features
- Train N **linear predictors**, one for each binary problem
- Each pivot predictor implicitly aligns non-pivot features from **source** & **target** domains

Penn
University of Pennsylvania

SCL: Dimensionality Reduction

$$\begin{bmatrix} | & & | & & | \\ \mathbf{w_1} & \cdots & \mathbf{w_i} & \cdots & \mathbf{w_N} \\ | & & | & & | \end{bmatrix}$$

- $\mathbf{W}^T\mathbf{x}$ gives N new features

- value of i^{th} feature is the propensity to see **"not buy"** in the same document

- **We still want fewer new features (1000 is too many)**
- **Many pivot predictors give similar information**
  - **"horrible", "terrible", "awful"**
- **Compute SVD & use top left singular vectors** Φ

Latent Semantic Indexing (LSI), (Deerwester et al. 1990)

Latent Dirichlet Allocation (LDA), (Blei et al. 2003)

Back to Linear Classifiers

$$\mathbf{x} = \begin{bmatrix} 0.3 \\ 0 \\ \vdots \\ 0 \\ 0.1 \end{bmatrix}$$

Classifier $\mathrm{sgn}\left[\mathbf{w}\cdot\mathbf{x} + \mathbf{v}\cdot\mathbf{\Phi}^T\mathbf{x}\right]$

- **Source** training: Learn $\mathbf{w}$ & $\mathbf{v}$ together

$$\mathbf{\Phi}^T\mathbf{x} = \begin{bmatrix} 0.3 \\ -1.0 \\ \vdots \\ 0.7 \\ -2.1 \end{bmatrix}$$

- **Target** testing: First apply $\mathbf{\Phi}$, then apply $\mathbf{w}$ and $\mathbf{v}$

Inspirations for SCL

1. **Alternating Structural Optimization (ASO)**
   - **Ando & Zhang** (JMLR 2005)
   - Inducing structures for semi-supervised learning

2. **Correspondence Dimensionality Reduction**
   - **Ham, Lee, & Saul** (AISTATS 2003)
   - Learn a low-dimensional representation from high-dimensional correspondences

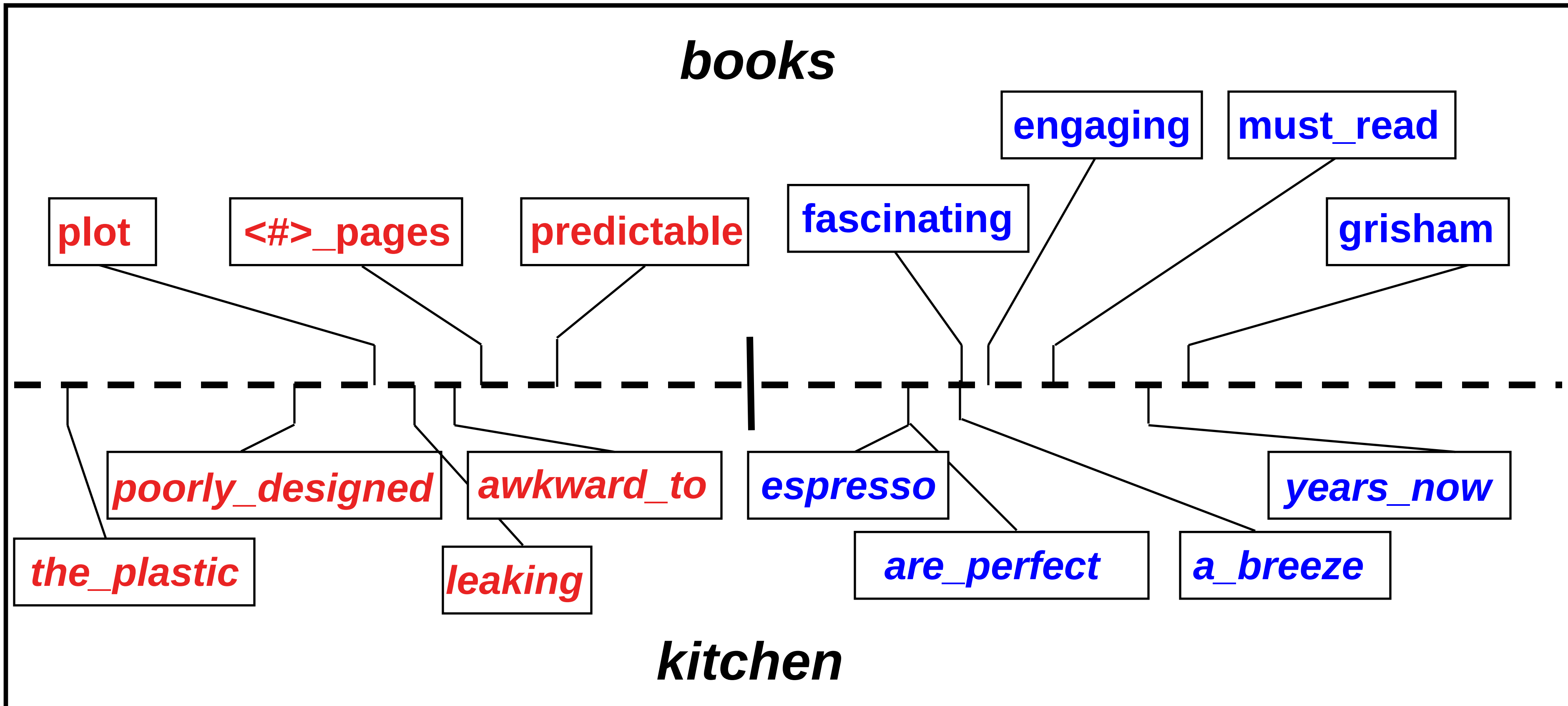
Sentiment Classification Data

- **Product reviews from Amazon.com**
  - Books, DVDs, Kitchen Appliances, Electronics
  - 2000 labeled reviews from each domain
  - 3000 – 6000 unlabeled reviews
- **Binary classification problem**
  - Positive if 4 stars or more, negative if 2 or fewer
- **Features:** unigrams & bigrams
- **Pivots:** SCL & SCL-MI
- **At train time:** minimize Huberized hinge loss (Zhang, 2004)

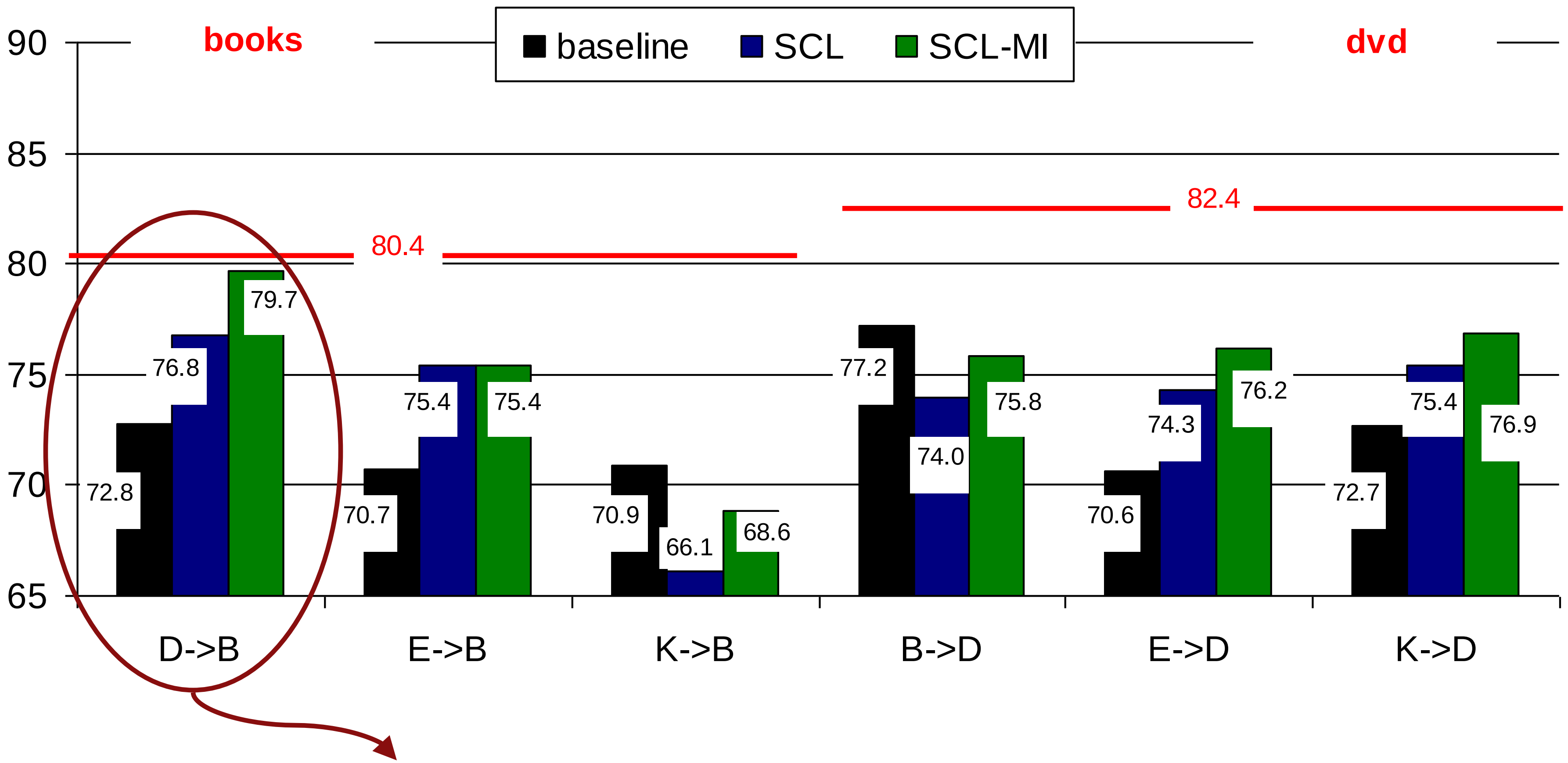
Penn
University of Pennsylvania

Visualizing Φ (books & kitchen)

negative **vs.** **positive**



Empirical Results: books & DVDs

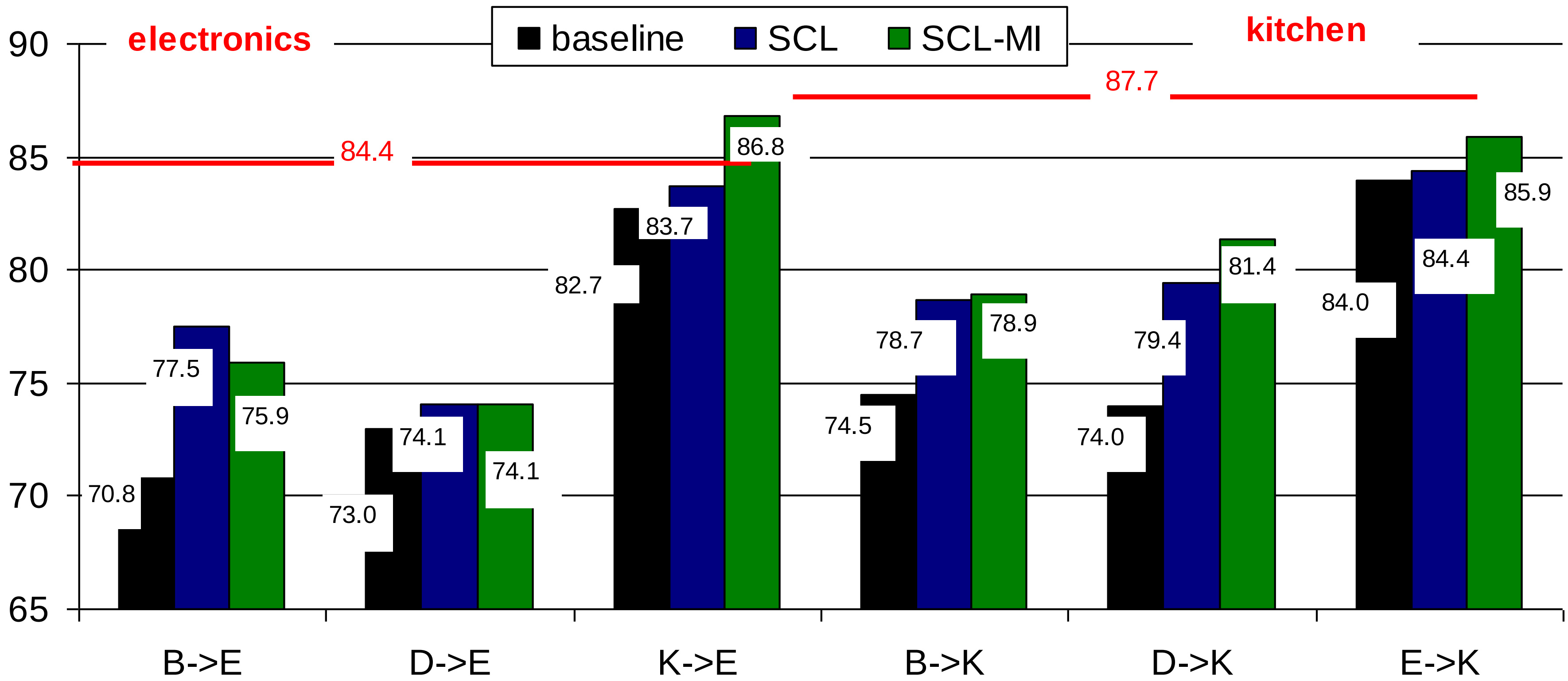


baseline loss due to adaptation: 7.6%

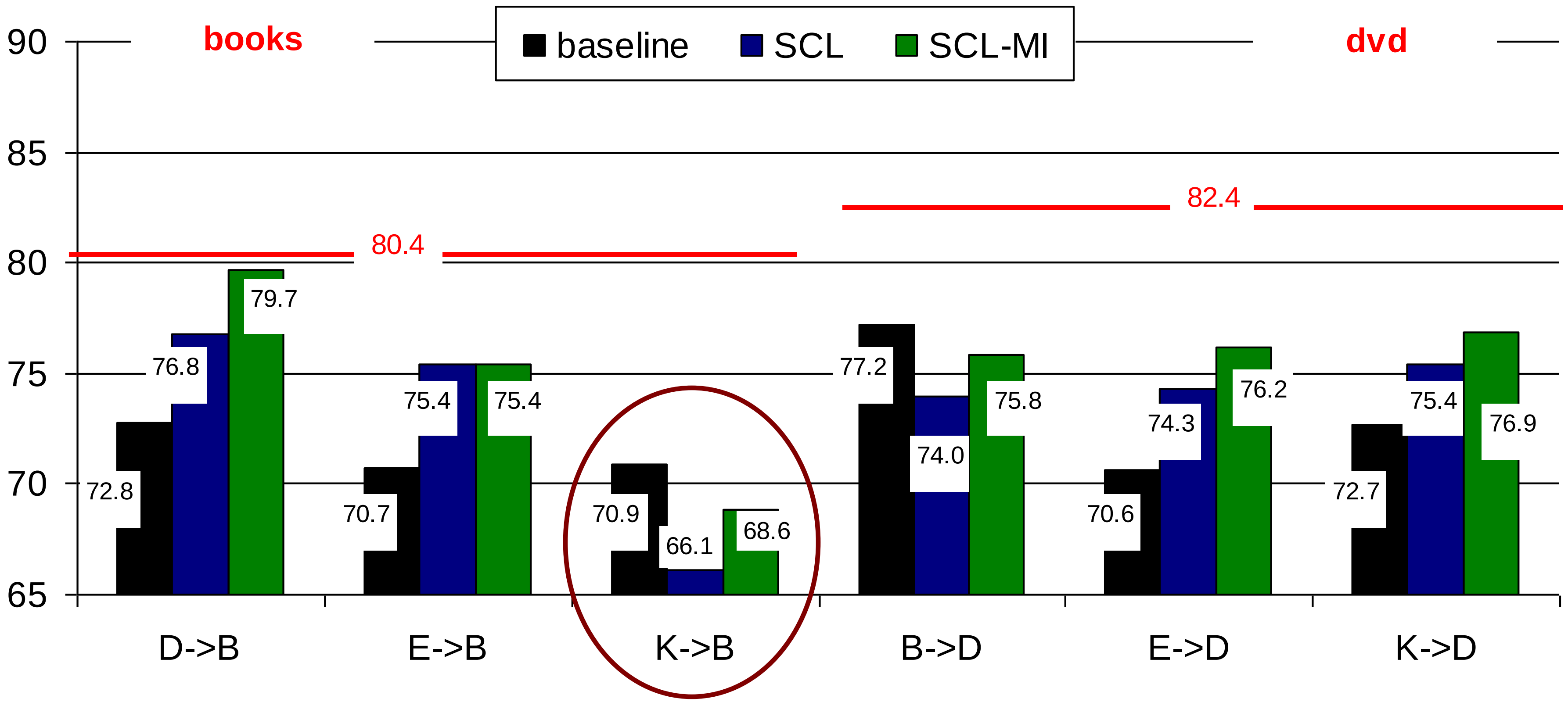
SCL-MI loss due to adaptation: 0.7%

University of Pennsylvania

Empirical Results: electronics & kitchen



Empirical Results: books & DVDs



- **Sometimes SCL can cause increases in error**
- **With only unlabeled data, we misalign features**

Penn
University of Pennsylvania

Using Labeled Data

50 instances of labeled target domain data

Source data, save weight vector for SCL features $\mathbf{v}_s$

Target data, regularize weight vector to be close to $\mathbf{v}_s$

Chelba & Acero, EMNLP 2004

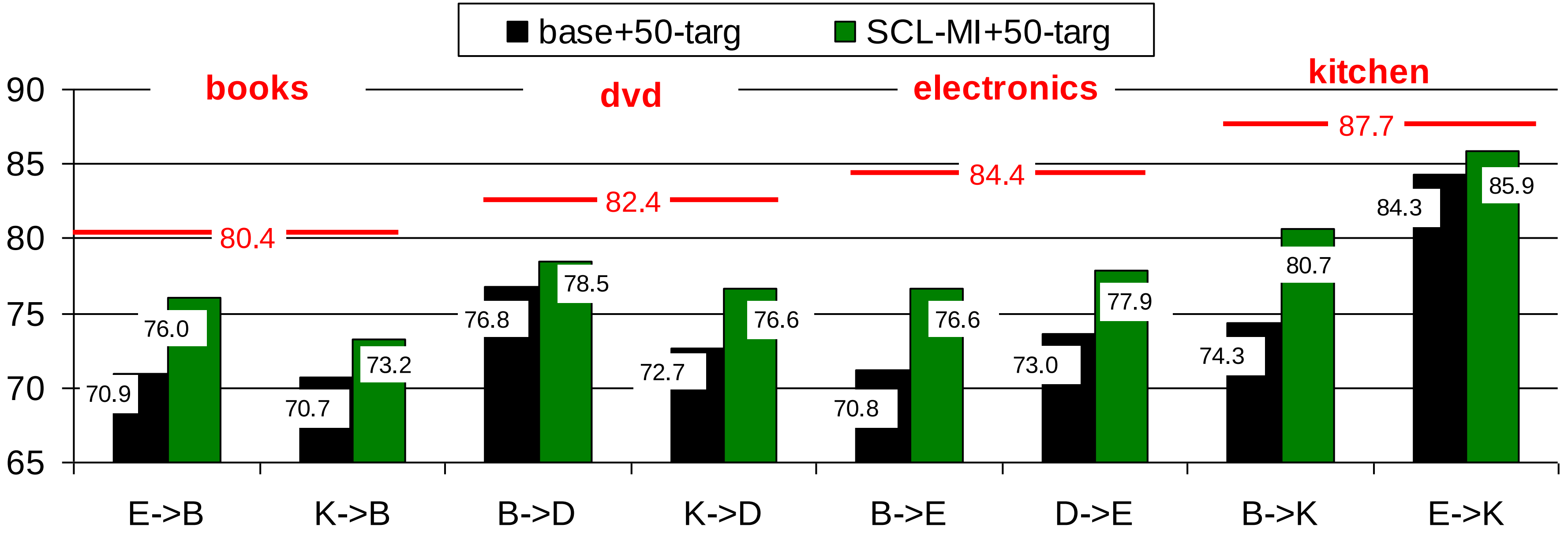
$$\min_{\mathbf{w},\mathbf{v}} \sum_j L\left(\mathbf{w}^T \mathbf{x}_j + \mathbf{v}^T \theta \mathbf{x}_j, y_j\right) +$$

$$\lambda ||\mathbf{w}||^2 + \mu ||\mathbf{v} - \mathbf{v}_s||^2$$

Huberized hinge loss **Keep SCL weights close to source weights**

Avoid using high-dimensional features

Empirical Results: labeled data



• With 50 labeled target instances, SCL-MI **always** improves over baseline

Average Improvements

model	base	base +targ	scl	scl-mi	scl-mi +targ
Avg Adaptation Loss	9.1	9.1	7.1	5.8	**4.9**

- **scl-mi reduces error due to transfer by 36%**

- **adding 50 instances [Chelba & Acero 2004] without SCL does not help**

- **scl-mi + targ reduces error due to transfer by 46%**

Penn
UNIVERSITY of PENNSYLVANIA

Error Bounds for Domain Adaptation

- Training and testing data are drawn from different distributions

- Exploit **unlabeled data** to give computable error bounds for domain adaptation

- Use these bounds in an **adaptation active learning** experiment

A Bound on the Adaptation Error

Let h be a binary hypothesis. If $\mathcal{E}$ is the set of measureable subsets of $\mathcal{X}$ and $\mathcal{D}_S, \mathcal{D}_T$ are source and target distributions with density functions p_S, p_T. Then

$$\begin{aligned} \epsilon_{\mathcal{D}_T}(h) &\leq \epsilon_{\mathcal{D}_S}(h) + \int |p_T(\mathbf{x}) - p_S(\mathbf{x})| \, d\mathbf{x} \\ &\leq \epsilon_{\mathcal{D}_S}(h) + 2 \sup_{E \in \mathcal{E}} \left| \Pr_{\mathcal{D}_T}[E] - \Pr_{\mathcal{D}_S}[E] \right| \end{aligned}$$

1. **Difference across all measurable subsets cannot be estimated from finite samples**

2. **We're only interested in differences related to classification error**

The $\mathcal{H}\Delta\mathcal{H}$ distance

Idea: Measure subsets where hypotheses in $\mathcal{H}$ disagree

Let $\mathcal{H}$ be a hypothesis class. Denote by $\mathcal{H}\Delta\mathcal{H}$ the set of subsets of $\mathcal{X}$ where two hypotheses in $\mathcal{H}$ disagree.

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2 \sup_{A \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathcal{D}_T}[A] - \Pr_{\mathcal{D}_S}[A] \right|$$

Subsets A are **error sets** of one hypothesis wrt another

1. Always lower than L_1
2. computable from finite **unlabeled** samples.
3. train classifier to discriminate between source and target data

For unlabeled samples $\mathcal{U}_S, U_T$, we write $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$

The optimal joint hypothesis

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \, \epsilon_{\mathcal{D}_S}(h) + \epsilon_{\mathcal{D}_T}(h)$$

$$\lambda = \epsilon_{\mathcal{D}_S}(h^*) + \epsilon_{\mathcal{D}_T}(h^*)$$

h^* is the hypothesis with **minimal combined error**

λ is that error

Penn
University of Pennsylvania

A Computable Adaptation Bound

Let $\mathcal{H}$ be a hypothesis class of VC dimension d and $\mathcal{U}_S, \mathcal{U}_T$ be unlabeled samples of size m' each, drawn from $\mathcal{D}_S, \mathcal{D}_T$ respectively. With probability at least $1-\delta$ (over the choice of unlabeled sample), for every $h \in \mathcal{H}$,

$$\epsilon_{\mathcal{D}_T}(h) \leq \epsilon_{\mathcal{D}_S}(h) + \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + \lambda + O\left(\sqrt{\frac{d \log \frac{m'}{d} + \log \frac{1}{\delta}}{m'}}\right)$$

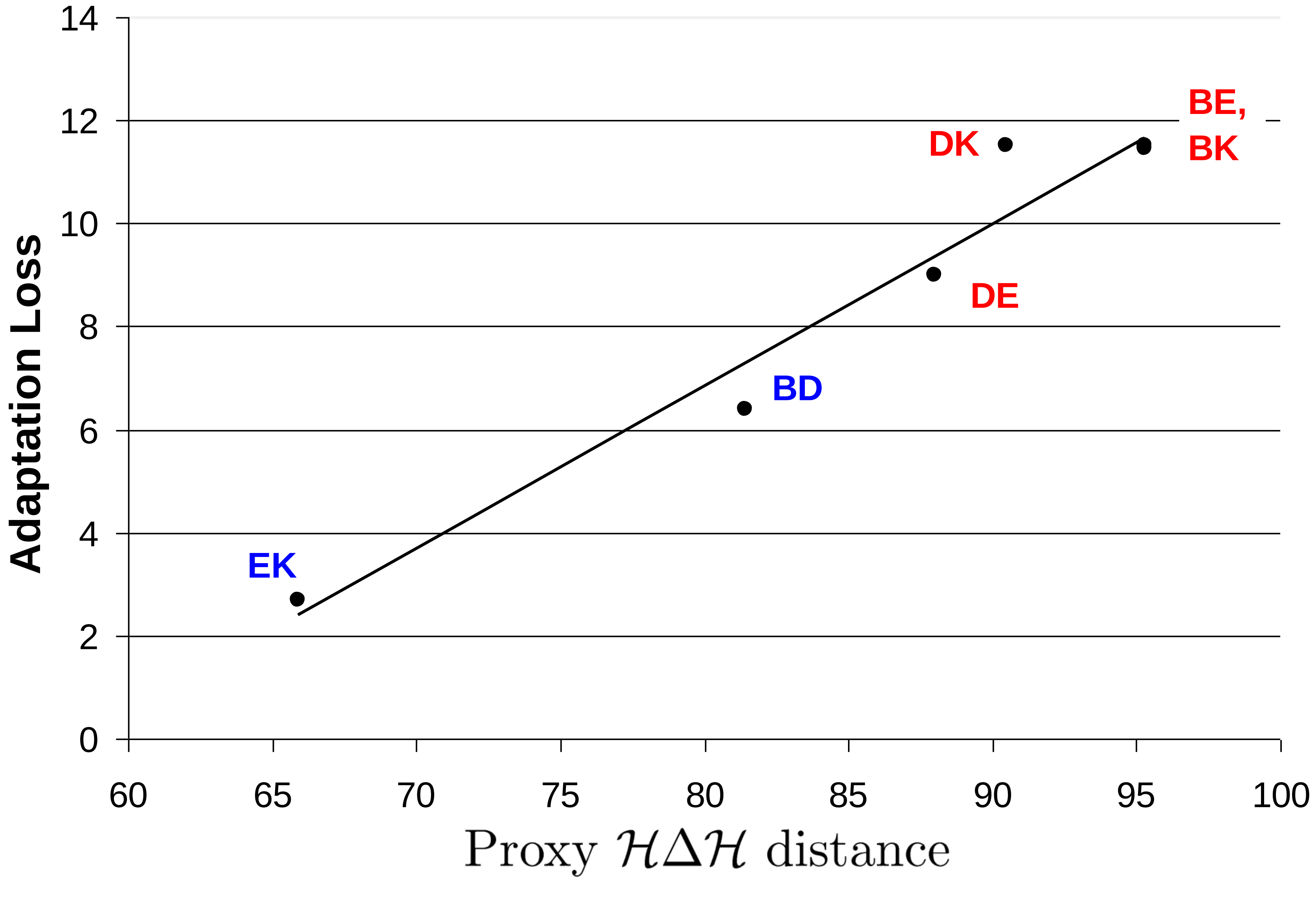
Divergence estimation complexity

Dependent on number of unlabeled samples

Adaptation Active Learning

- **Given limited resources, which domains should we label?**
- **Train a classifier to distinguish between unlabeled source and target instances**
- **Proxy $\mathcal{H}\Delta\mathcal{H}$ - distance: classifier margin**
- **Label domains to get the most coverage**
  - **one of (books, DVDs)**
  - **one of (electronics, kitchen)**

Proxy $\mathcal{H}\Delta\mathcal{H}$ distance: Train a linear classifier to distinguish between unlabeled instances from two domains



Adaptation & Ranking

- **Input: query & list of top-ranked documents**
- **Output: Ranking**
- **Score documents based on editorial or click-through data**
- **Adaptation: Different markets or query types**
- **Pivots: common relevant features**

Advertisement: More SCL & Theory

Domain Adaptation with Structural Correspondence Learning.

John Blitzer, Ryan McDonald, Fernando Pereira.

EMNLP 2006.

Learning Bounds for Domain Adaptation.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, Jenn Wortman.

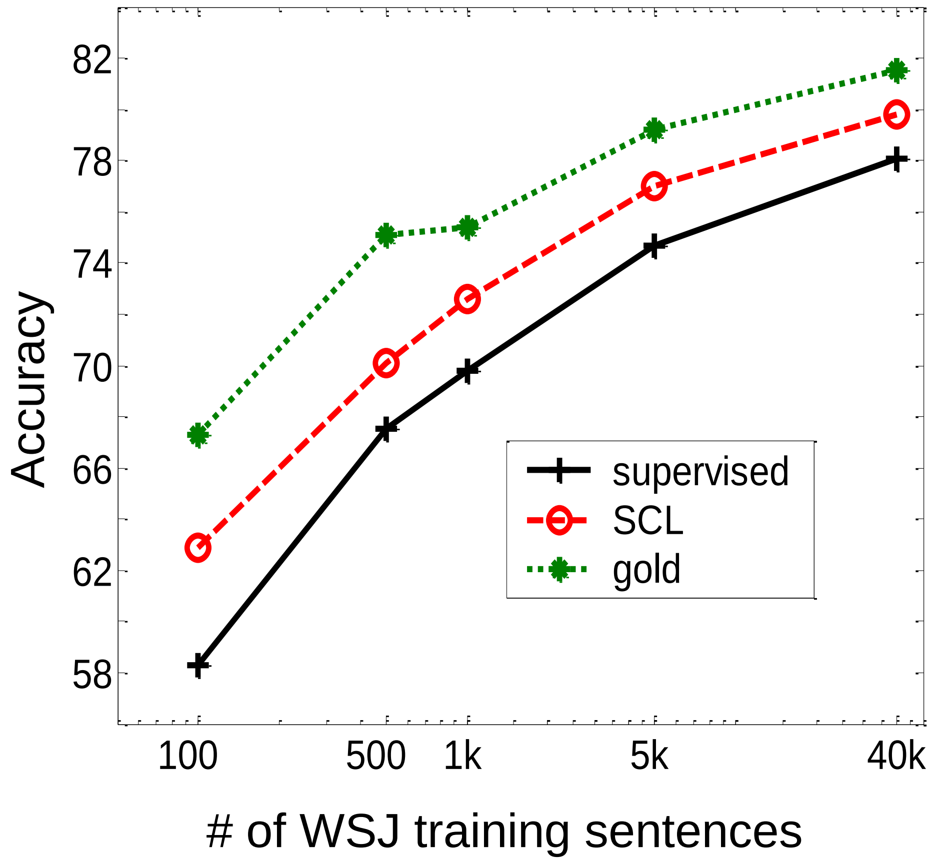
Currently under review.

Pipeline Adaptation: Tagging & Parsing

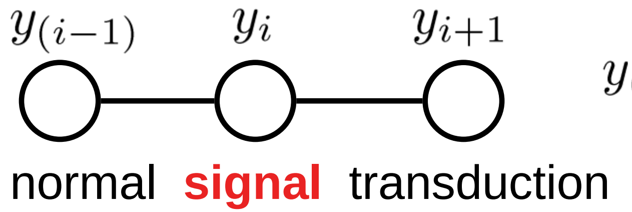
Dependency Parsing

- McDonald et al. 2005
- Uses part of speech tags as features
- Train on WSJ, test on MEDLINE
- Use different taggers for MEDLINE input features

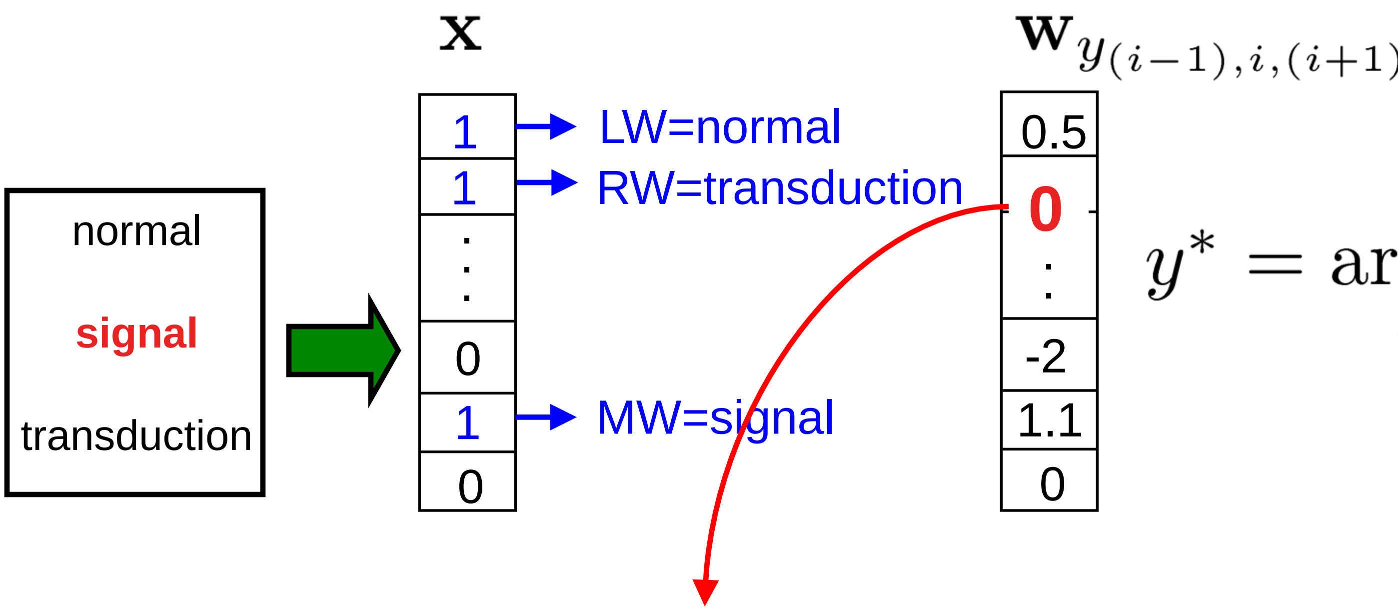
Accuracy for different tagger inputs



Features & Linear Models



$$y_{(i-1),i,(i+1)} = \text{JJ-NN-NN}$$



$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} \left[\mathbf{w}_y \cdot \mathbf{x} \right]$$

Problem: If we've only trained on financial news, then **w(RW=transduction) = 0**

Future Work

- **SCL for other problems & modalities**
  - **named entity recognition**
  - **vision (aligning SIFT features)**
  - **speaker / acoustic environment adaptation**
- **Learning low-dimensional representations for multi-part prediction problems**
  - **natural language parsing, machine translation, sentence compression**

Penn
UNIVERSITY of PENNSYLVANIA

Learning Bounds for Adaptation

- **Standard learning bound, binary classification**

Let $\mathcal{H}$ be a hypothesis class of VC dimension d. If we draw m samples $\mathbf{x} \in \mathcal{S}$ from $\mathcal{D}$ and label them according to $f : \mathcal{X} \to [0,1]$, then with probability $1-\delta$, for every $h \in \mathcal{H}$,

$$\epsilon_{\mathcal{D}}(h,f) \leq \hat{\epsilon}_{\mathcal{S}}(h,f) + O\left(\sqrt{\frac{d \log \frac{m}{d} + \log \frac{1}{\delta}}{m}}\right)$$

- **Target data is drawn from a different distribution than source data**