A Two-Stage Approach to Domain Adaptation for Statistical Classifiers

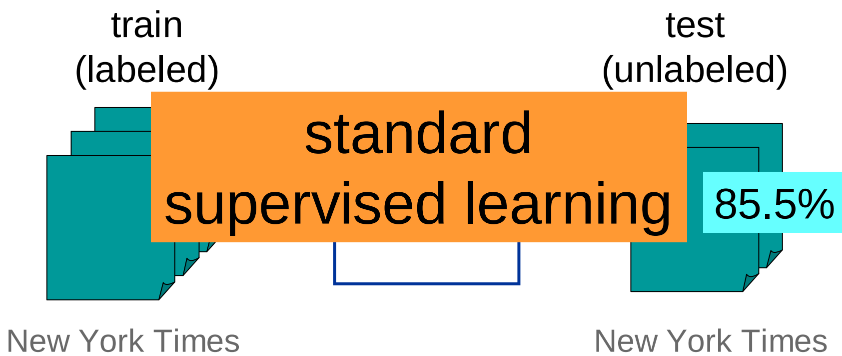
Jing Jiang & ChengXiang Zhai

Department of Computer Science
University of Illinois at Urbana-Champaign

What is domain adaptation?

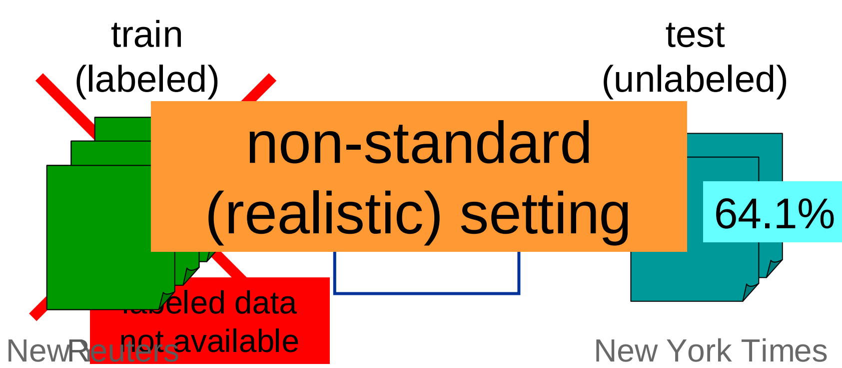
Example: named entity recognition

persons, locations, organizations, etc.

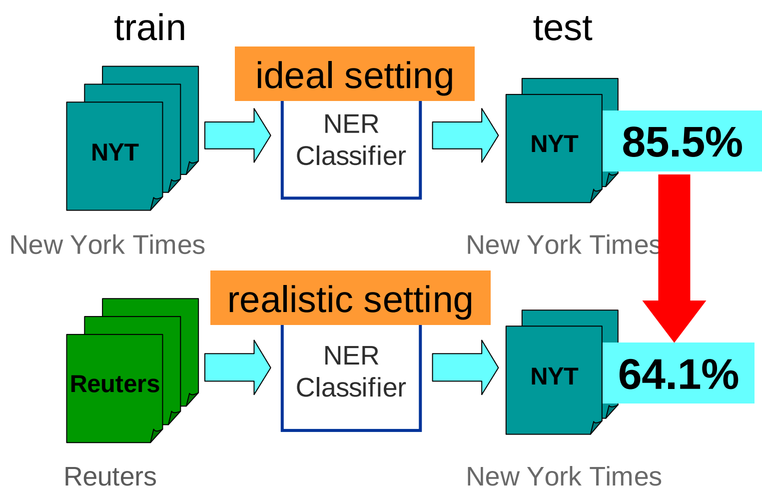


Example: named entity recognition

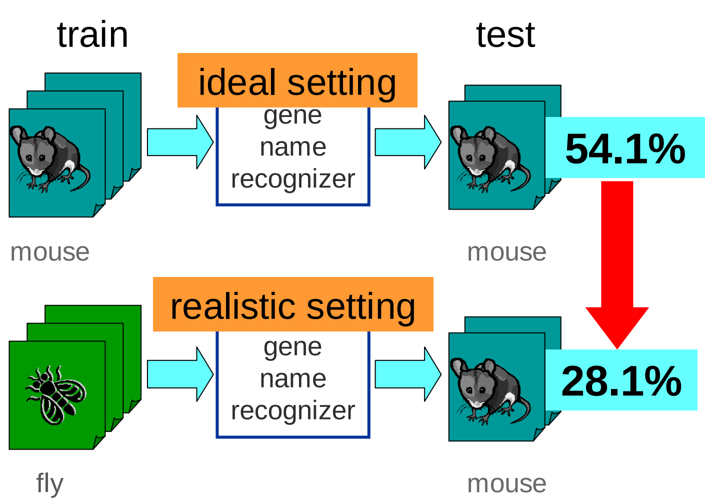
persons, locations, organizations, etc.



Domain difference → performance drop



Another NER example



Other examples

- Spam filtering:
  - Public email collection → personal inboxes
- Sentiment analysis of product reviews
  - Digital cameras → cell phones
  - Movies → books
- Can we do better than standard supervised learning?
- Domain adaptation: to design learning methods that are aware of the training and test domain difference.

How do we solve the problem in general?

Observation 1

domain-specific features

wingless
daughterless
eyeless
apexless
…

Observation 1

domain-specific features

wingless
daughterless
eyeless
apexless
…

- describing phenotype
- in fly gene nomenclature
- feature **"-less"** weighted high

feature still useful for other organisms?

No!

CD38
PABPC5
…

Observation 2

generalizable features

...**decapentaplegic** and **wingless** are expressed in analogous patterns in each...

...that **CD38** is expressed by both neurons and glial cells...that **PABPC5** is expressed in fetal brain and in a range of adult tissues.

Observation 2

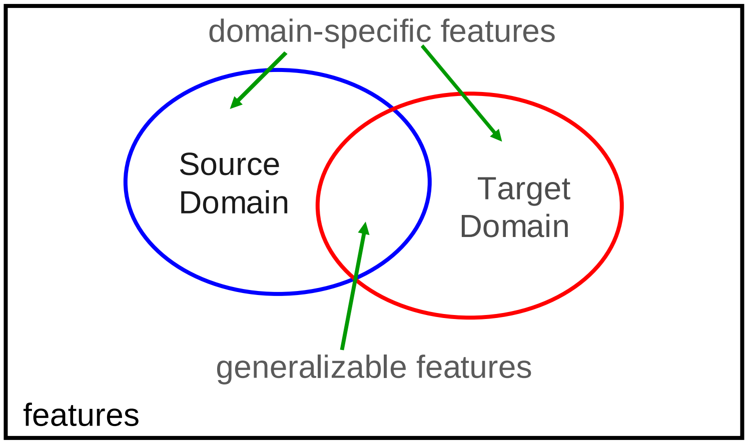
generalizable features

...decapentaplegic and wingless are **expressed** in analogous patterns in each...

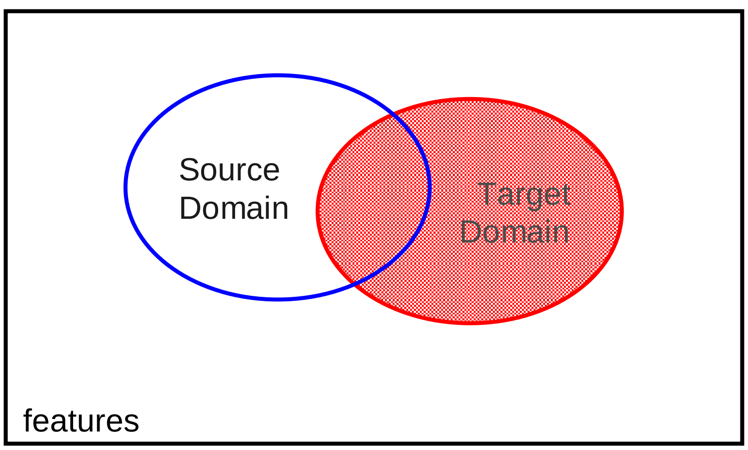
...that CD38 is **expressed** by both neurons and glial cells...that PABPC5 is **expressed** in fetal brain and in a range of adult tissues.

feature “X be expressed”

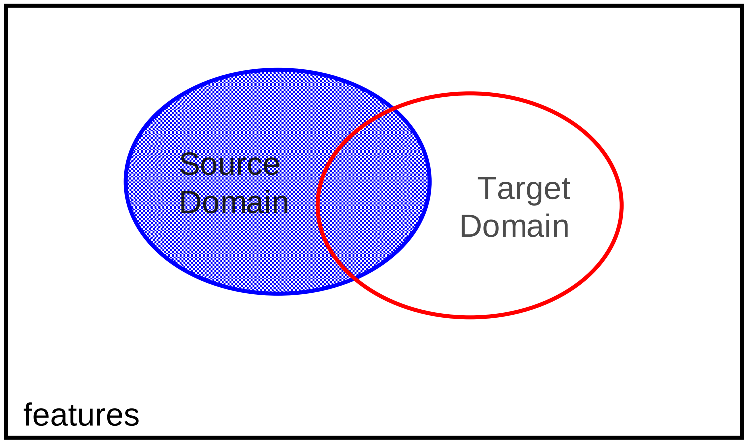
General idea: two-stage approach



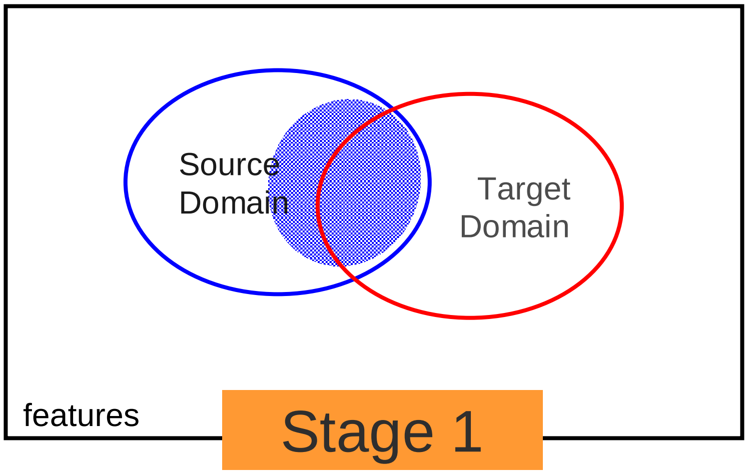
Goal



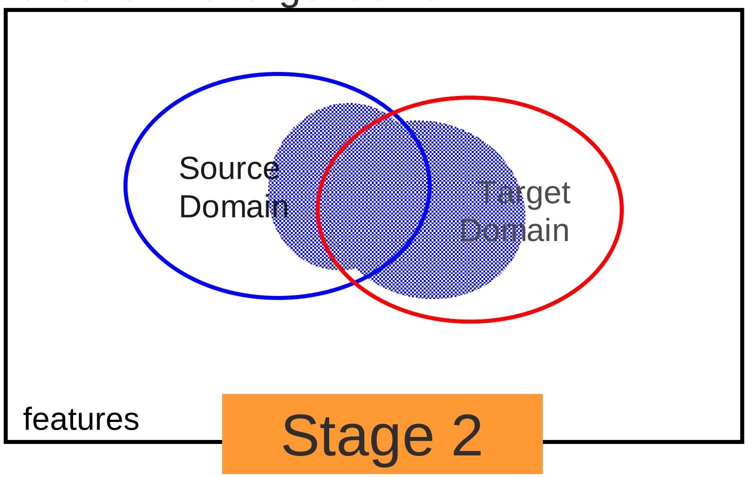
Regular classification



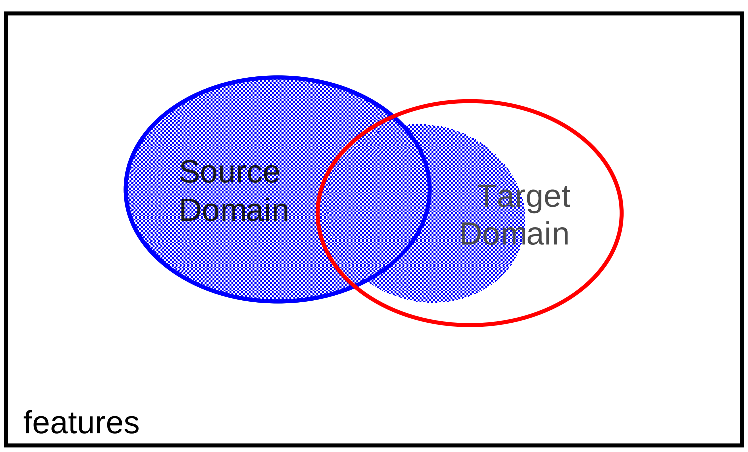
Generalization: to emphasize generalizable features in the trained model



Adaptation: to pick up domain-specific features for the target domain



Regular semi-supervised learning



Comparison with related work

- We explicitly model generalizable features.
  - Previous work models it implicitly [Blitzer et al. 2006, Ben-David et al. 2007, Daumé III 2007].
- We do not need labeled target data but we need multiple source (training) domains.
  - Some work requires labeled target data [Daumé III 2007].
- We have a 2^{nd} stage of adaptation, which uses semi-supervised learning.
  - Previous work does not incorporate semi-supervised learning [Blitzer et al. 2006, Ben-David et al. 2007, Daumé III 2007].

Implementation of the two-stage approach with logistic regression classifiers

Logistic regression classifiers

$$\begin{pmatrix} 0.2 \\ 4.5 \\ 5 \\ -0.3 \\ 3.0 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

-less

p binary features

X be expressed

… and **wingless** are expressed in…

$$p(y \mid \mathbf{x}, \mathbf{w}) = \frac{\exp(\mathbf{w}_y^T \mathbf{x})}{\sum_{y'} \exp(\mathbf{w}_{y'}^T \mathbf{x})}$$

$\mathbf{w}_y \mathbf{w}_y^T \mathbf{x} \ \mathbf{x}$

Learning a logistic regression classifier

$$\begin{pmatrix} 0.2 \\ 4.5 \\ 5 \\ -0.3 \\ 3.0 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

regularization term

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left(\lambda \|\mathbf{w}\|^2 \right.$$

$N \quad i=1 \quad \sum_{y'} \exp(\mathbf{w}_{y'} \mathbf{x})$

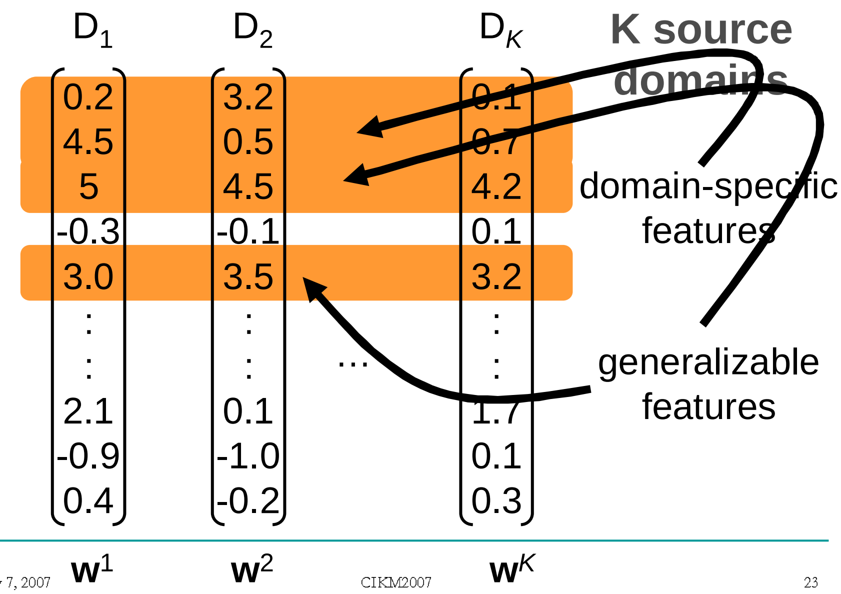
penalize large weights

control model complexity

log likelihood of training data

$\mathbf{w}_y^T \mathbf{x}$

Generalizable features in weight vectors



We want to decompose **w** in this way

$$\begin{bmatrix} 0.2 \\ 4.5 \\ 5 \\ -0.3 \\ 3.0 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4.6 \\ 0 \\ 3.2 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.2 \\ 4.5 \\ 0.4 \\ -0.3 \\ -0.2 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{bmatrix}$$

h non-zero entries for *h* generalizable features

Feature selection matrix A

matrix A selects h generalizable features

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ & & & \vdots & & & \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow h\left\{ \begin{pmatrix} 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \right.$$

$$\mathbf{z} = A\mathbf{x}$$

Decomposition of **w**

weights for domain-specific features

weights for generalizable features

$$\begin{pmatrix} 0.2 \\ 4.5 \\ 5 \\ -0.3 \\ 3.0 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4.6 \\ 3.2 \\ \vdots \\ \vdots \\ 3.6 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 4.5 \\ 0.4 \\ -0.3 \\ -0.2 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\mathbf{w}^T \mathbf{x} = \mathbf{v}^T \mathbf{z} + \mathbf{u}^T \mathbf{x}$$

Decomposition of **w**

$$\mathbf{w}^T\mathbf{x} = \mathbf{v}^T\mathbf{z} + \mathbf{u}^T\mathbf{x}$$
$$= \mathbf{v}^T A\mathbf{x} + \mathbf{u}^T\mathbf{x}$$
$$= (A\mathbf{v})^T\mathbf{x} + \mathbf{u}^T\mathbf{x}$$

$$\boxed{\mathbf{w} = A^T\,\mathbf{v} + \mathbf{u}}$$

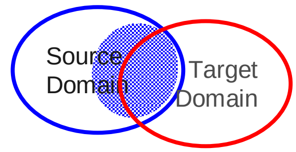
Decomposition of **w**

shared by al domain-specific

$$\begin{bmatrix} 0.2 \\ 4.5 \\ 5 \\ -0.3 \\ 3.0 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{bmatrix} = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \vdots & \\ & & \vdots & \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \end{pmatrix} \boxed{\begin{bmatrix} 4.6 \\ 3.2 \\ \vdots \\ \vdots \\ 3.6 \end{bmatrix}} + \boxed{\begin{bmatrix} 0.2 \\ 4.5 \\ 0.4 \\ -0.3 \\ -0.2 \\ \vdots \\ \vdots \\ 2.1 \\ -0.9 \\ 0.4 \end{bmatrix}}$$

$$\mathbf{w} \quad = \quad A^T\,\mathbf{v} \quad + \quad \mathbf{u}$$

Framework for generalization



Fix *A*, optimize:

$$(\hat{\mathbf{v}},\{\hat{\mathbf{u}}^k\}) = \arg\min_{\mathbf{v},\{\mathbf{u}^k\}} \Bigg[\lambda\Big(\|\mathbf{v}\|^2 + \lambda_s \sum_{k=1}^{K} \|\mathbf{u}^k\|^2 \Big) - \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{i=1}^{N_k} \log p(y_i^k \mid \mathbf{x}_i^k; A^T\mathbf{v}+\mathbf{u}^k) \Bigg]$$

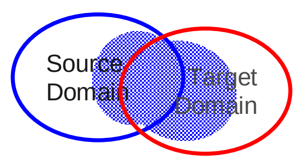
regularization term

likelihood of labeled data from K source domains

$\lambda_s >> 1$: to penalize domain-specific features

$\mathbf{w}^k$

Framework for adaptation



Fix *A*, optimize:

$$(\hat{\mathbf{v}},\hat{\mathbf{u}}^t,\{\hat{\mathbf{u}}^k\}) = \arg\min_{\mathbf{v},\mathbf{u}^t\{\mathbf{u}^k\}} \Bigg[\lambda\Big(\|\mathbf{v}\|^2 + \lambda_s \sum_{k=1}^{K} \|\mathbf{u}^k\|^2 + \lambda_t \|\mathbf{u}^t\|^2 \Big) - \frac{1}{K+1}\Big(\sum_{k=1}^{N_k}\frac{1}{N_k}\sum_{i=1}^{N_k} \log p(y_i^k \mid \mathbf{x}_i^k; A^T\mathbf{v}+\mathbf{u}^k) + \frac{1}{m}\sum_{i=1}^{m} \log p(y_i^t \mid \mathbf{x}_i^t; A^T\mathbf{v}+\mathbf{u}^t) \Big) \Bigg]$$

likelihood of pseudo labeled target-domain examples

$\lambda_t = 1 << \lambda_s$: to pick up domain-specific features in the target domain

How to find A? (1)

- Joint optimization

$$(\hat{A}, \hat{\mathbf{v}}, \{\hat{\mathbf{u}}^k\}) = \arg\min_{A, \mathbf{v}, \{\mathbf{u}^k\}} \left[\lambda \left(\|\mathbf{v}\|^2 + \lambda_s \sum_{k=1}^{K} \|\mathbf{u}^k\|^2 \right) - \frac{1}{K} \sum_{k=1}^{N_k} \frac{1}{N_k} \sum_{i=1}^{N_k} \log p(y_i^k \mid \mathbf{x}_i^k; A^T \mathbf{v} + \mathbf{u}^k) \right]$$

  - Alternating optimization

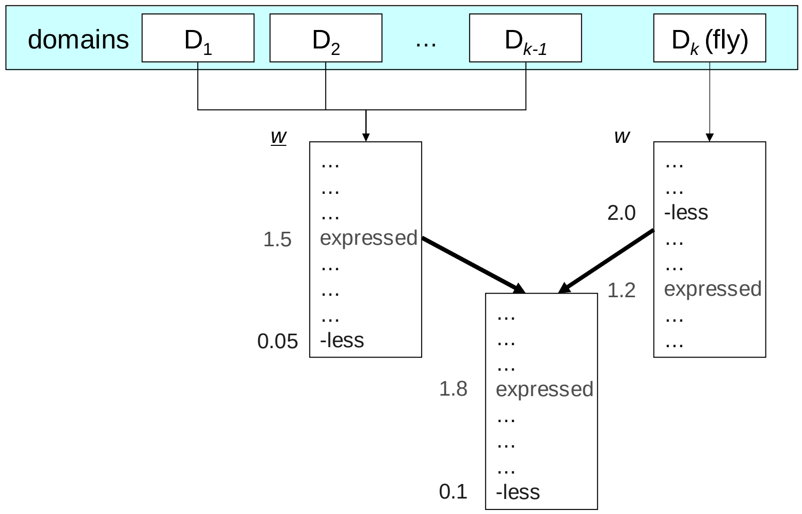
How to find A? (2)

- Domain cross validation
  - Idea: training on (K – 1) source domains and test on the held-out source domain
  - Approximation:
    - w_f^k: weight for feature f learned from domain k
    - $\underline{w}_f^k$: weight for feature f learned from other domains
    - rank features by

$$\sum_{k=1}^{K} w_f^k \cdot \underline{w}_f^k$$

  - See paper for details

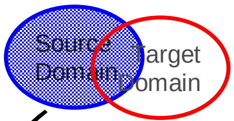
Intuition for domain cross validation



Experiments

- Data set
  - BioCreative Challenge Task 1B
  - Gene/protein name recognition
  - 3 organisms/domains: fly, mouse and yeast
- Experiment setup
  - 2 organisms for training, 1 for testing
  - F1 as performance measure

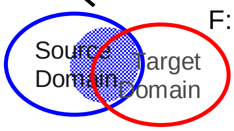
Experiments: Generalization



using generalizable features is effective

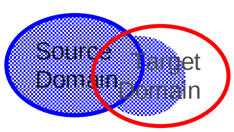
Method	F+M→Y	M+Y→F	Y+F→M
BL	0.633	0.129	0.416
DA-1 (joint-opt)	0.627	0.153	0.425
DA-2 (domain CV)	**0.654**	**0.195**	**0.470**

F: fly M: mouse Y: yeast



domain cross validation is more effective than joint optimization

Experiments: Adaptation

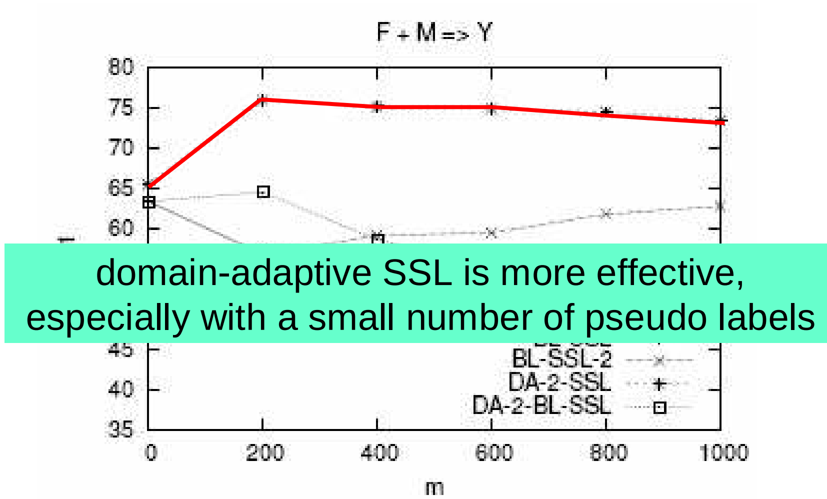


Method	F+M→Y	M+Y→F	Y+F→M
BL-SSL	0.633	0.241	0.458
DA-2-SSL	**0.759**	**0.305**	**0.501**

F: fly M: mouse Y: yeast

domain-adaptive bootstrapping is more effective than regular bootstrapping

Experiments: Adaptation



Conclusions and future work

- Two-stage domain adaptation
  - Generalization: outperformed standard supervised learning
  - Adaptation: outperformed standard bootstrapping
- Two ways to find generalizable features
  - Domain cross validation is more effective
- Future work
  - Single source domain?
  - Setting parameters *h* and *m*

References

- S. Ben-David, J. Blitzer, K. Crammer & F. Pereira. *Analysis of representations for domain adaptation*. NIPS 2007.
- J. Blitzer, R. McDonald & F. Pereira. *Domain adaptation with structural correspondence learning*. EMNLP 2006.
- H. Daumé III. *Frustratingly easy domain adaptation*. ACL 2007.

Thank you!