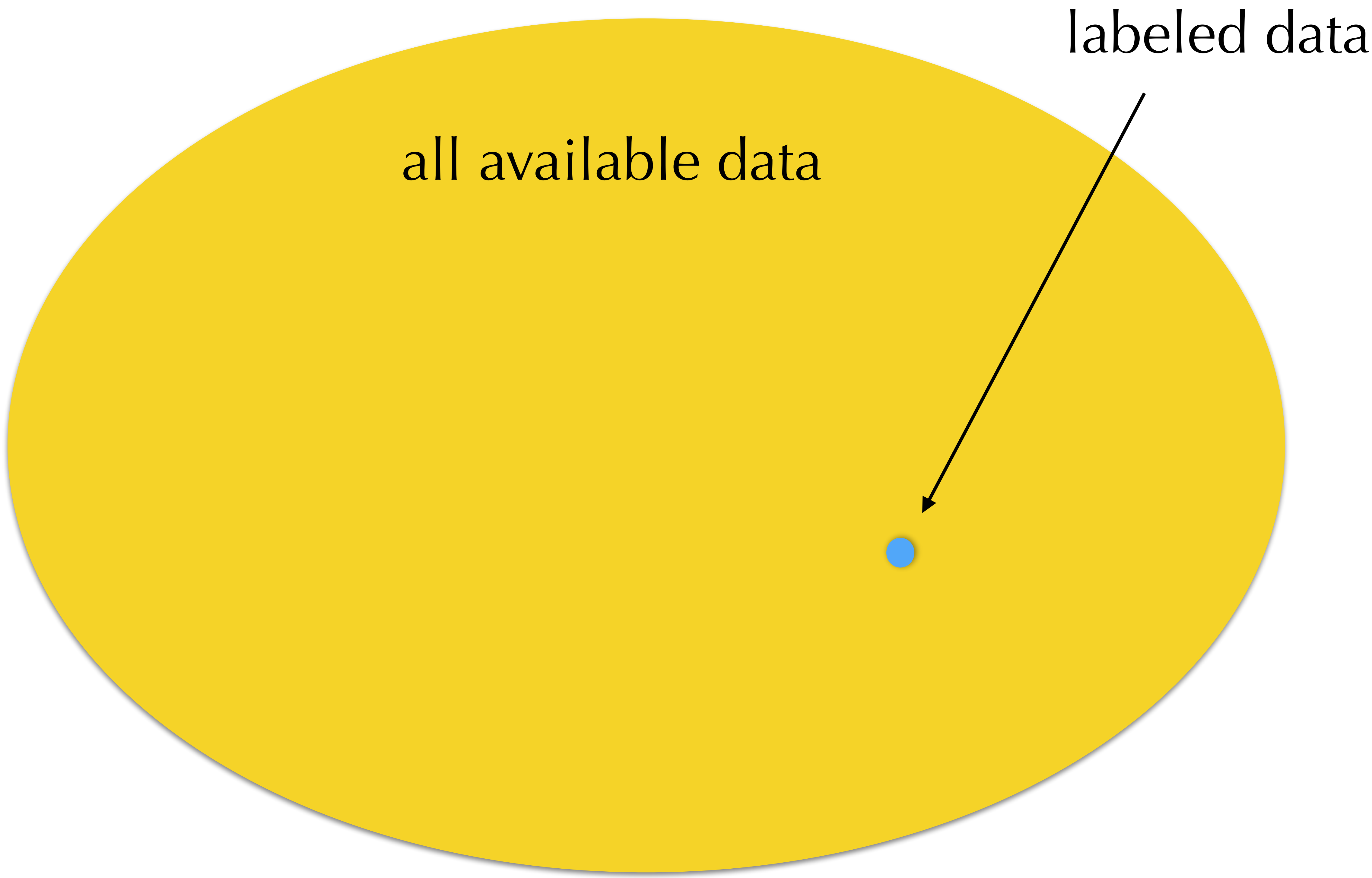
Benefits of HPC for NLP *besides big data*

Barbara Plank
Center for Sprogteknologie (CST)
University of Copenhagen, Denmark
http://cst.dk/bplank

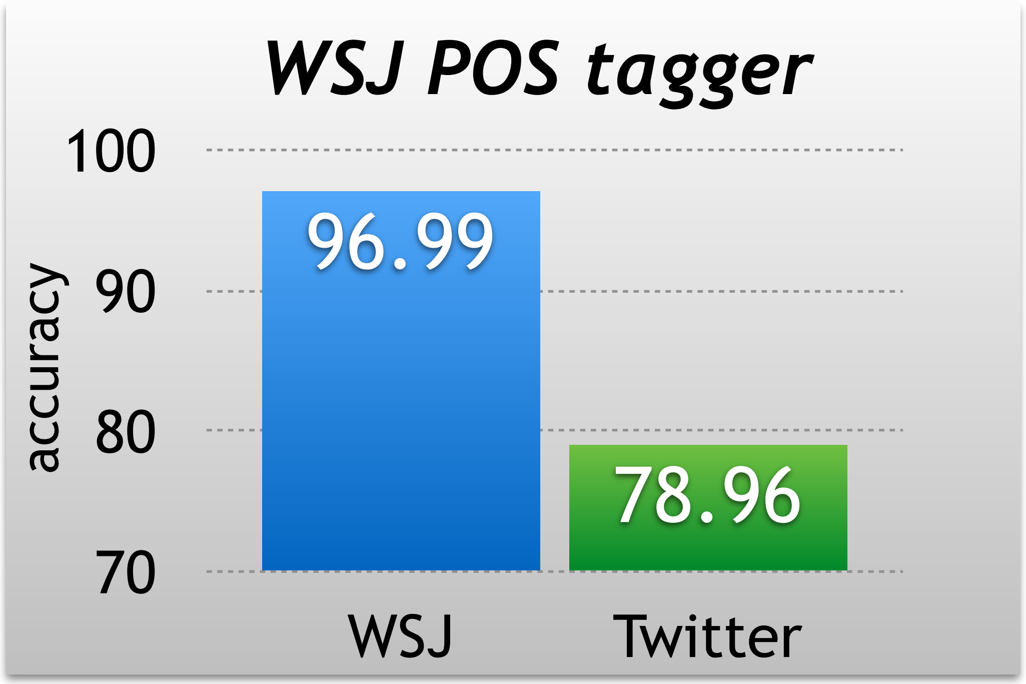
Web-Scale Natural Language Processing in Northern Europe
Oslo, Nov 24, 2014

Motivation

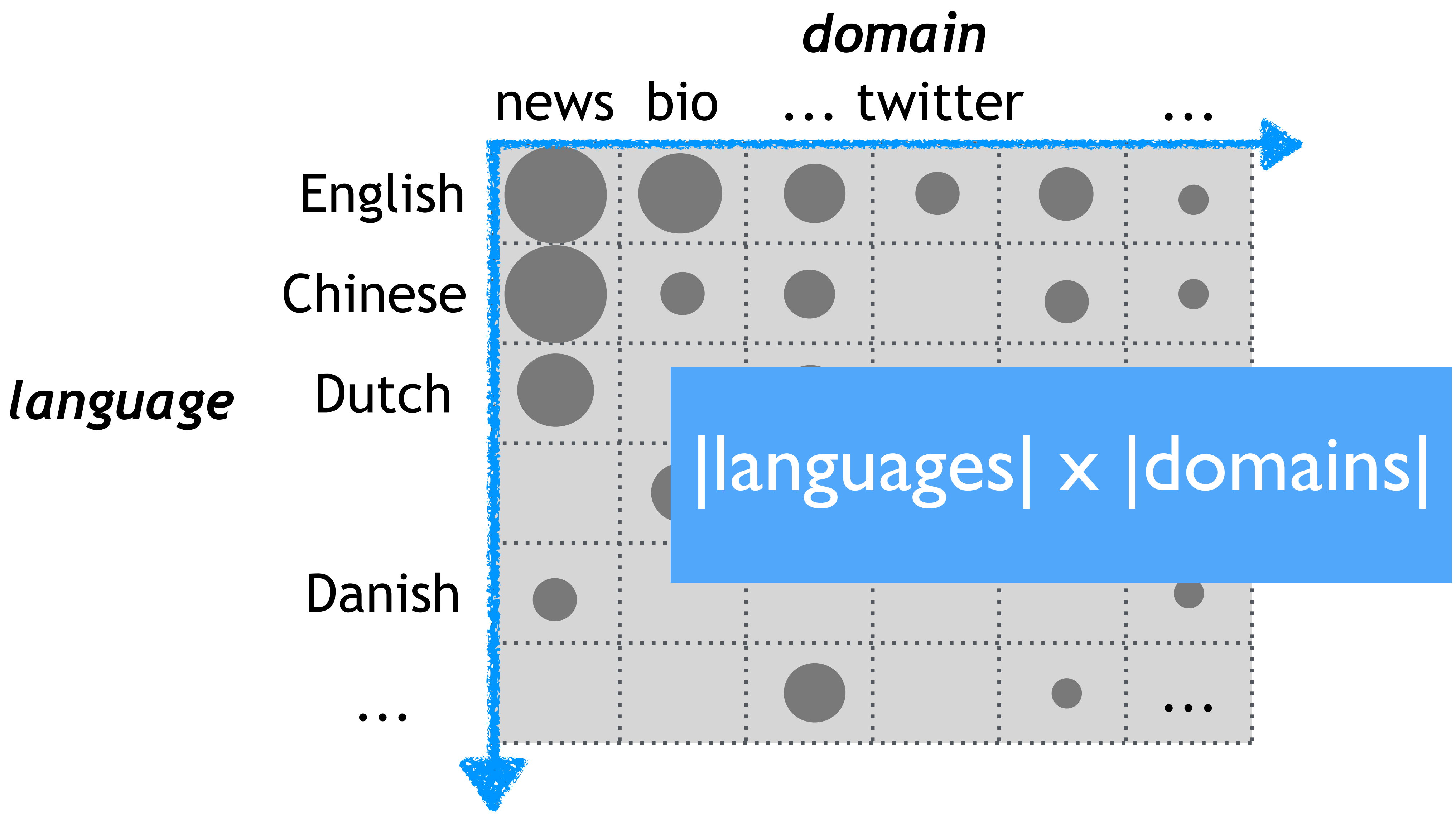


The problem: Training data is biased

the CROSS-DOMAIN GULF



The problem: Training data is scarce



Goal: Robust processing

- Exploit unlabeled data to improve NLP across **domains** and across **languages**

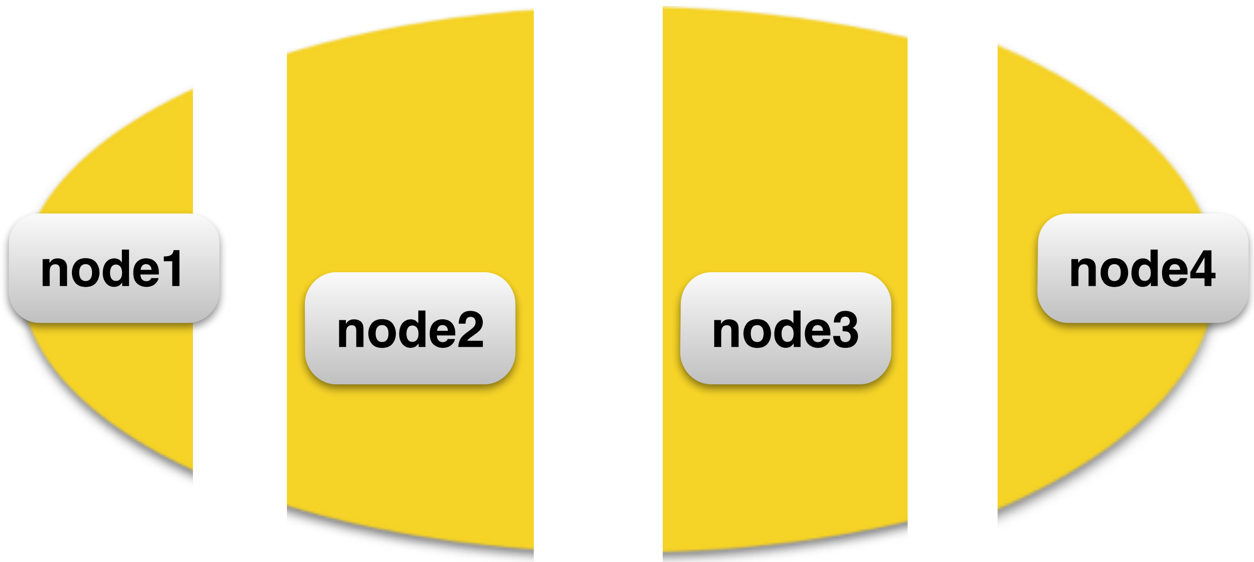
- Possible methods:

  - **unsupervised domain adaptation** (e.g. exploiting unlabeled data clustering/embeddings, importance weighting)

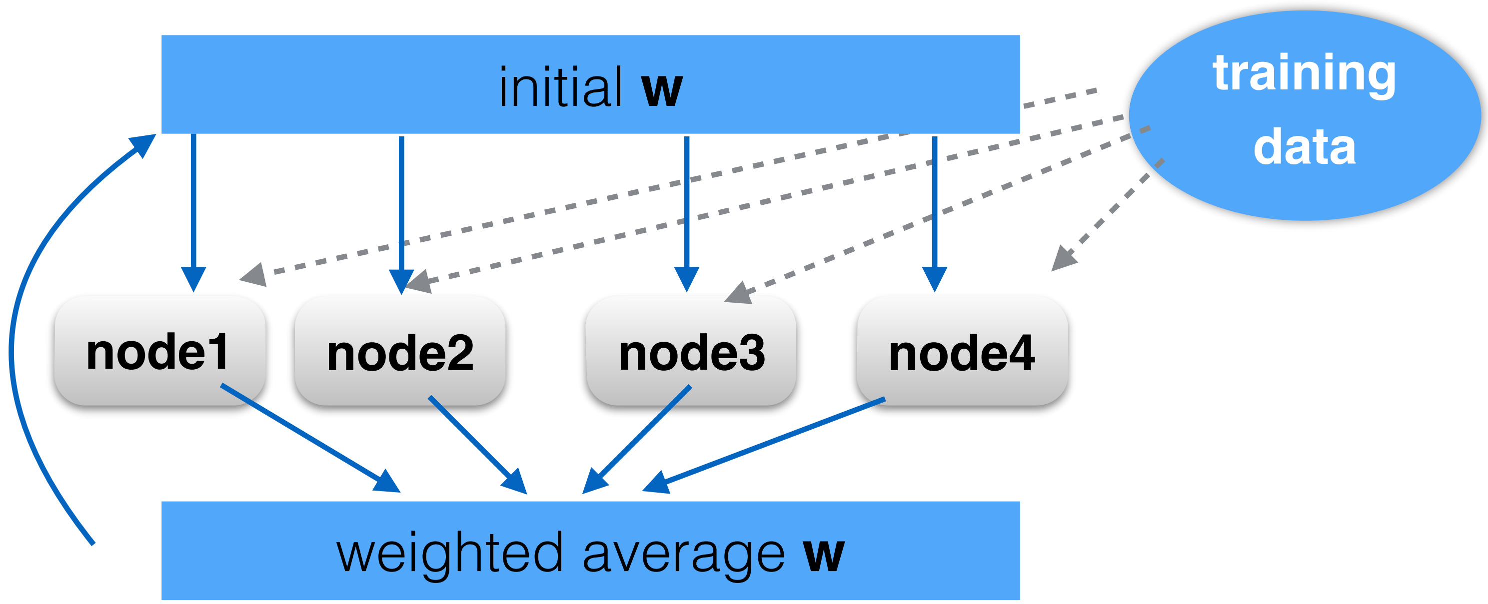
  - cross-lingual learning (not today's talk, just started)

Traditional HPC use in NLP

- Parallelize data processing



- Distributed model training (e.g. McDonald et al.,2010; Gesmundo & Tomeh, 2012)



Additional benefits of HPC

not only lots of unlabeled data…

unsupervised & semi-supervised algorithms

models: many parameters

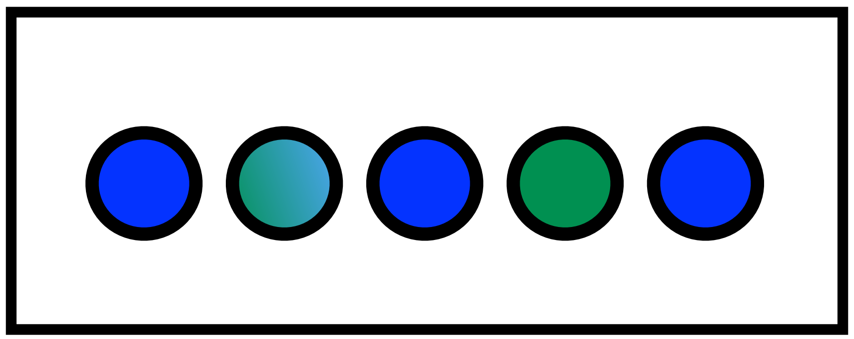
evaluation: need robust results

sharing: common data repositories

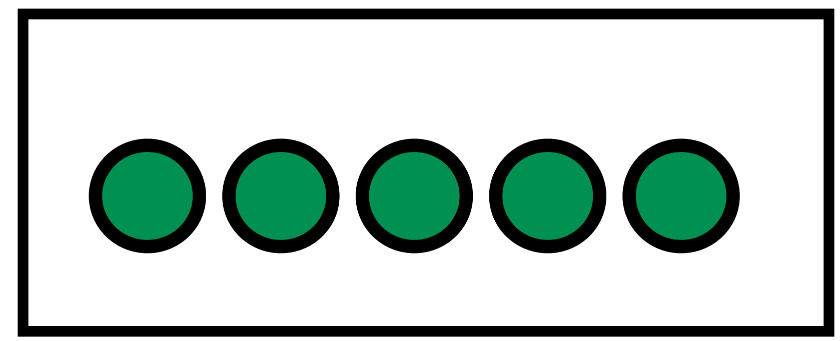
Example study: Importance weighting

Does importance weighting work for unsupervised DA of POS taggers?

SOURCE train



TARGET test



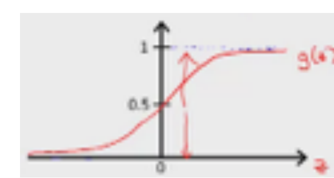
assign instance-dependent weights (Shimodaira, 2001):

?

$$\frac{P_T(\mathbf{x})}{P_S(\mathbf{x})}$$

unlabeled
TARGET

approximation, e.g.:

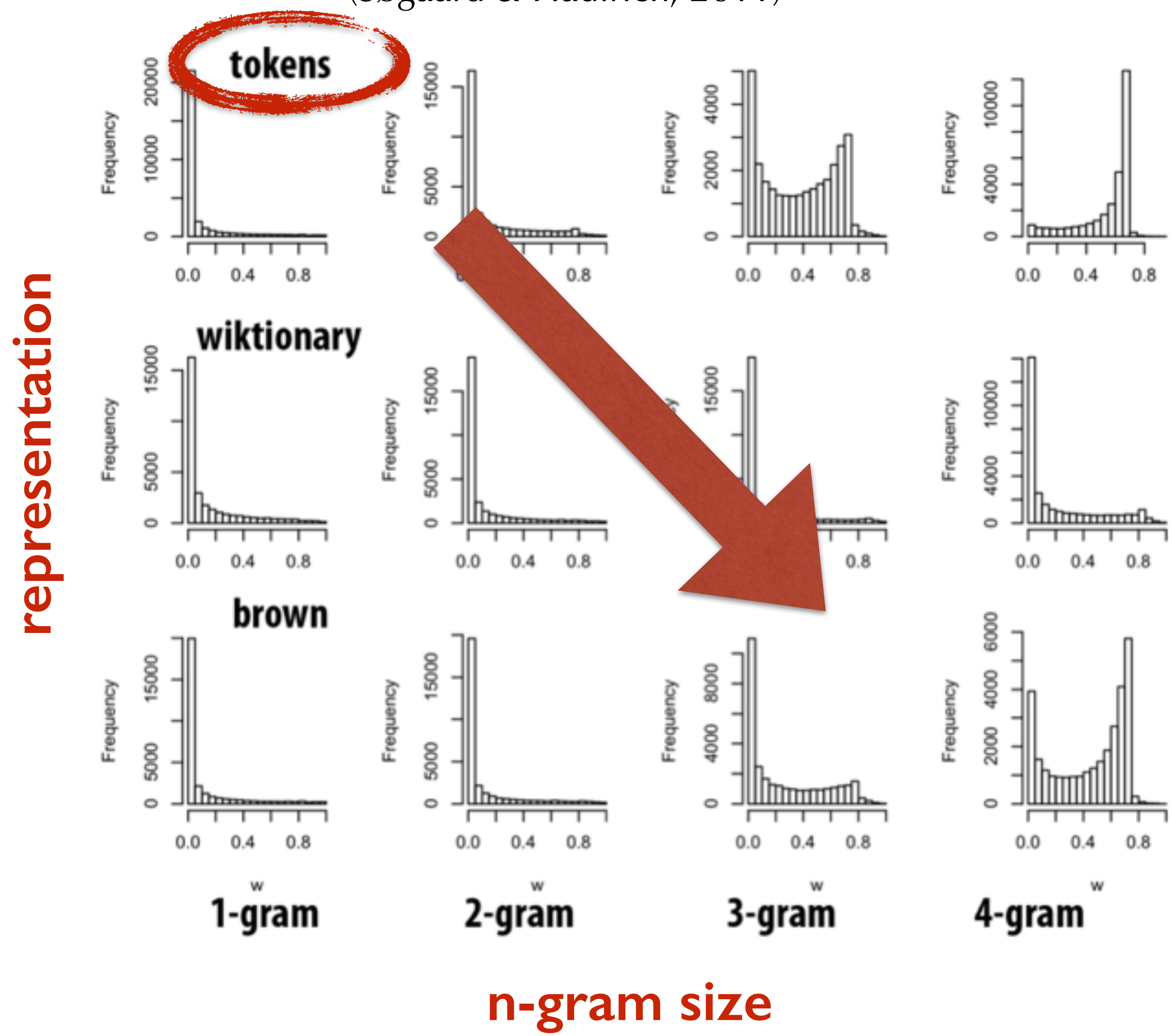


domain classifier to discriminate between SOURCE & TARGET

(Zadrozny et al., 2004; Bickel and Scheffer, 2007; Søgaard and Haulrich, 2011)

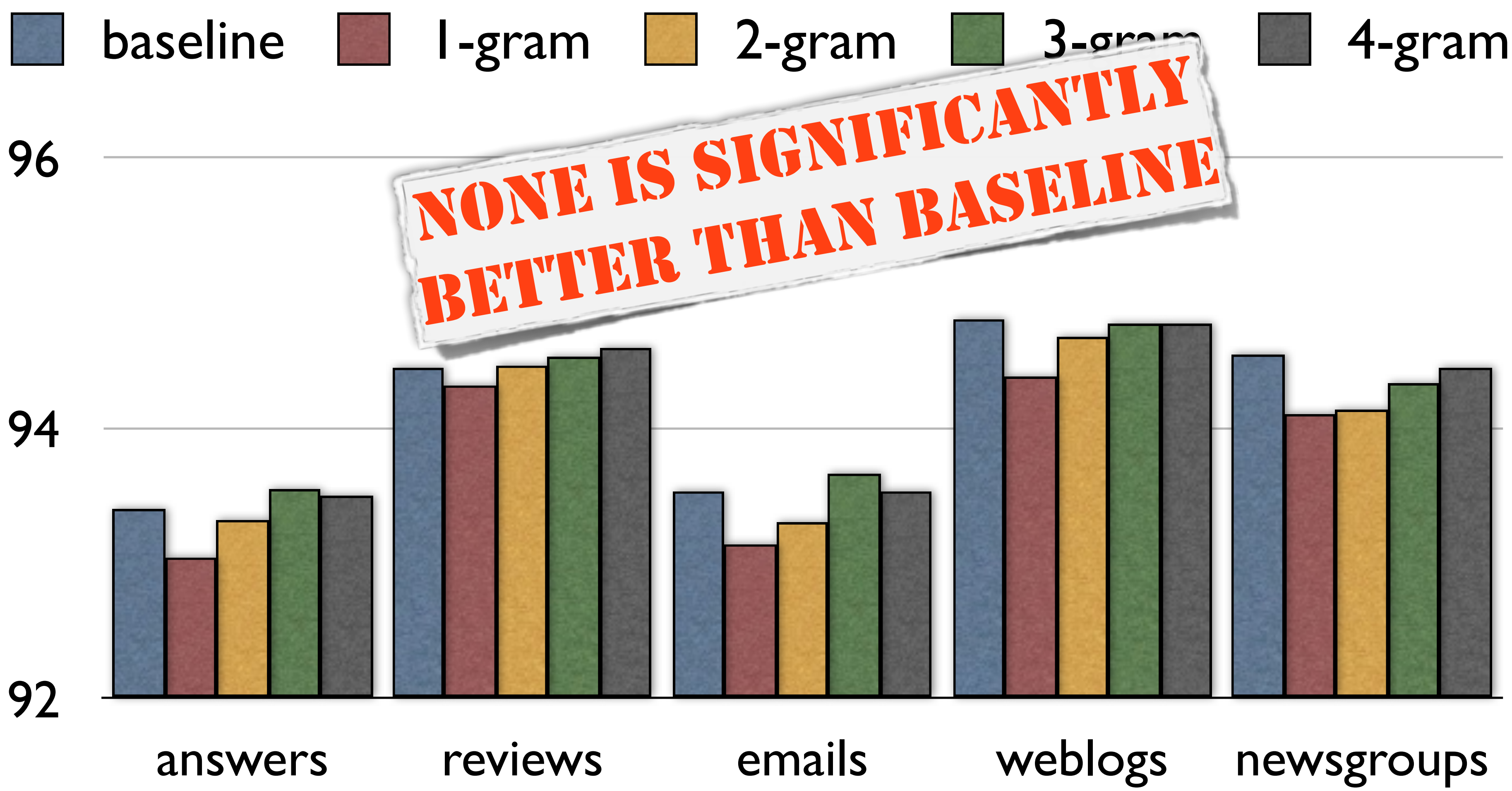
Domain classifier

(Søgaard & Haulrich, 2011)



Results

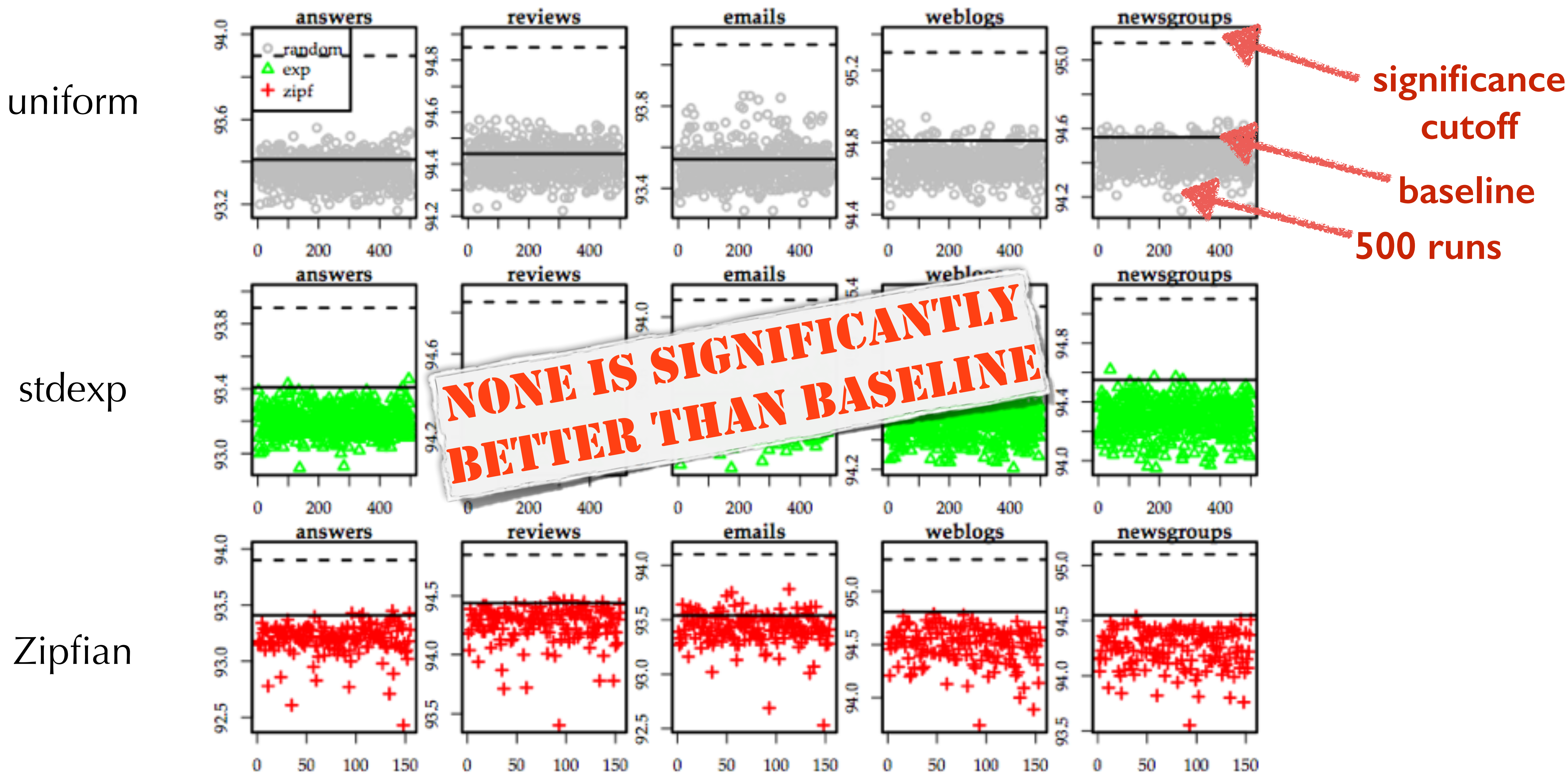
Token-based domain classifier



on test sets; results were similar for other representations (Brown, Wiktionary)

(Plank, Johannsen, Søgaard, 2014) EMNLP

Random weighting

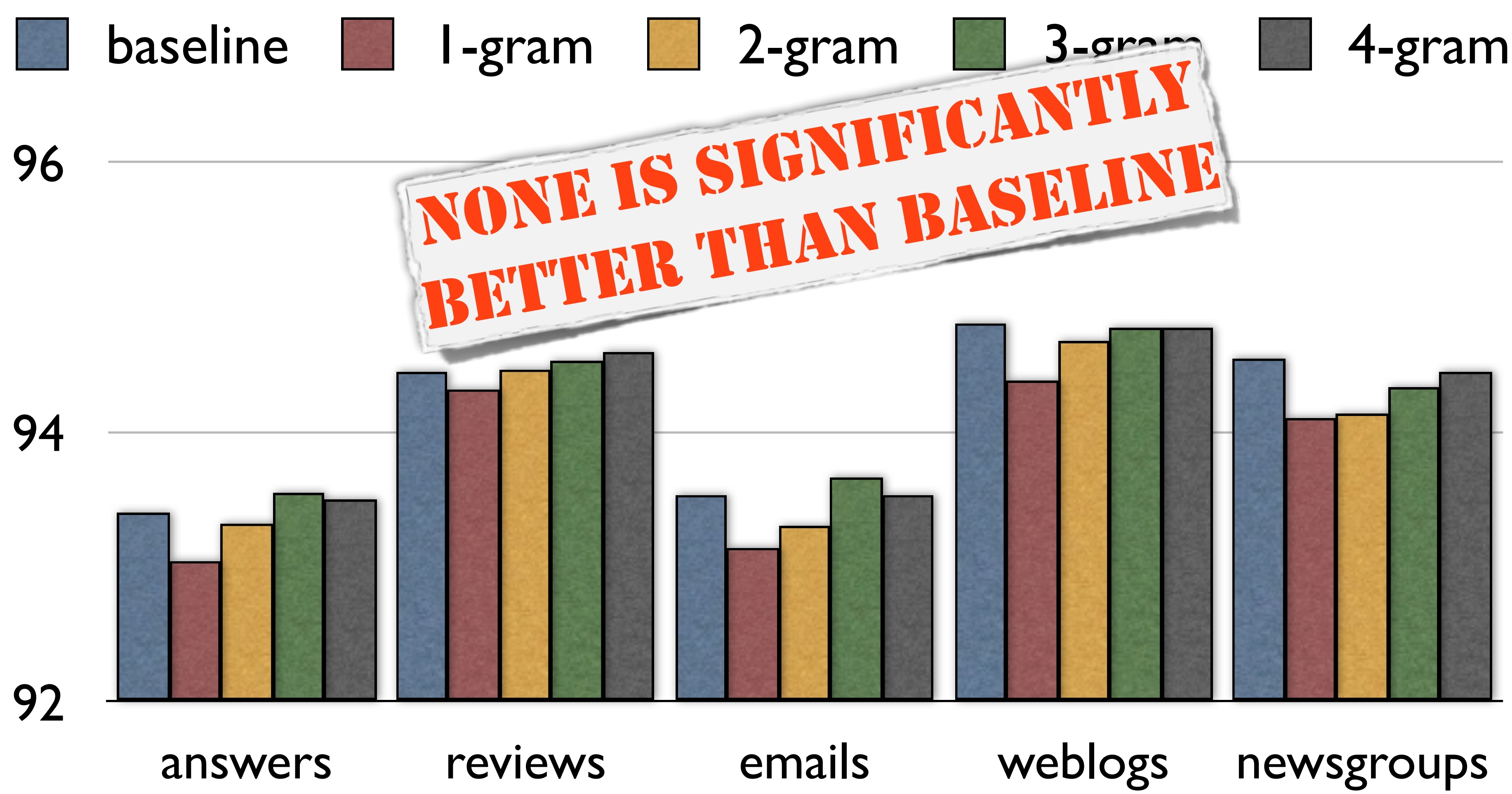


Each plot: 500 POS tagging models; total: 1500 models;
sequential: ~5m/model (7500m, **5 1/2 days**)
parallelized on HPC **Gardar** (in 3 batches) : **1.5 days**

(Plank, Johannsen, Søgaard, 2014) EMNLP

Results

Token-based domain classifier



	answers	reviews	emails	weblogs	newsgroups	
avg tag ambiguity	1.09	1.07	1.07	1.05	1.05	**low**
KL-div:	0.05	0.04	0.03	0.01	0.01	**low**
OOV:	27.7	29.5	29.9	22.1	23.1	**high OOV!**

on test sets; results were similar for other representations (Brown, Wiktionary)

Gardar

- we used the joint HPC cluster in Iceland for these experiments
- 288 nodes, 6 cores (= 3456 cores) 24GB each
- batch jobs submitted via TORQUE
- we have access since 6 months (end of April 2014): **Gardar is very useful!**
- we have locally only: 1 server with 8 cores, 384gb memory, 1.5TB disk space

evaluation

How robust are our results?

Within sample bias

- Twitter POS tagger, large differences on different Twitter samples:

	train/test	Gimpel	Ritter
Twitter data sets	Gimpel	90.46	82.29
	Ritter	80.52	90.40
	Combined	89.19	87.43

(Hovy et al., LREC 2014; Fromheide et al., 2014)

What to do about this?

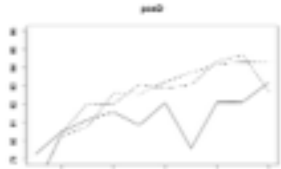
- Whenever possible **evaluate**:
  - across several test data sets

A	B	C

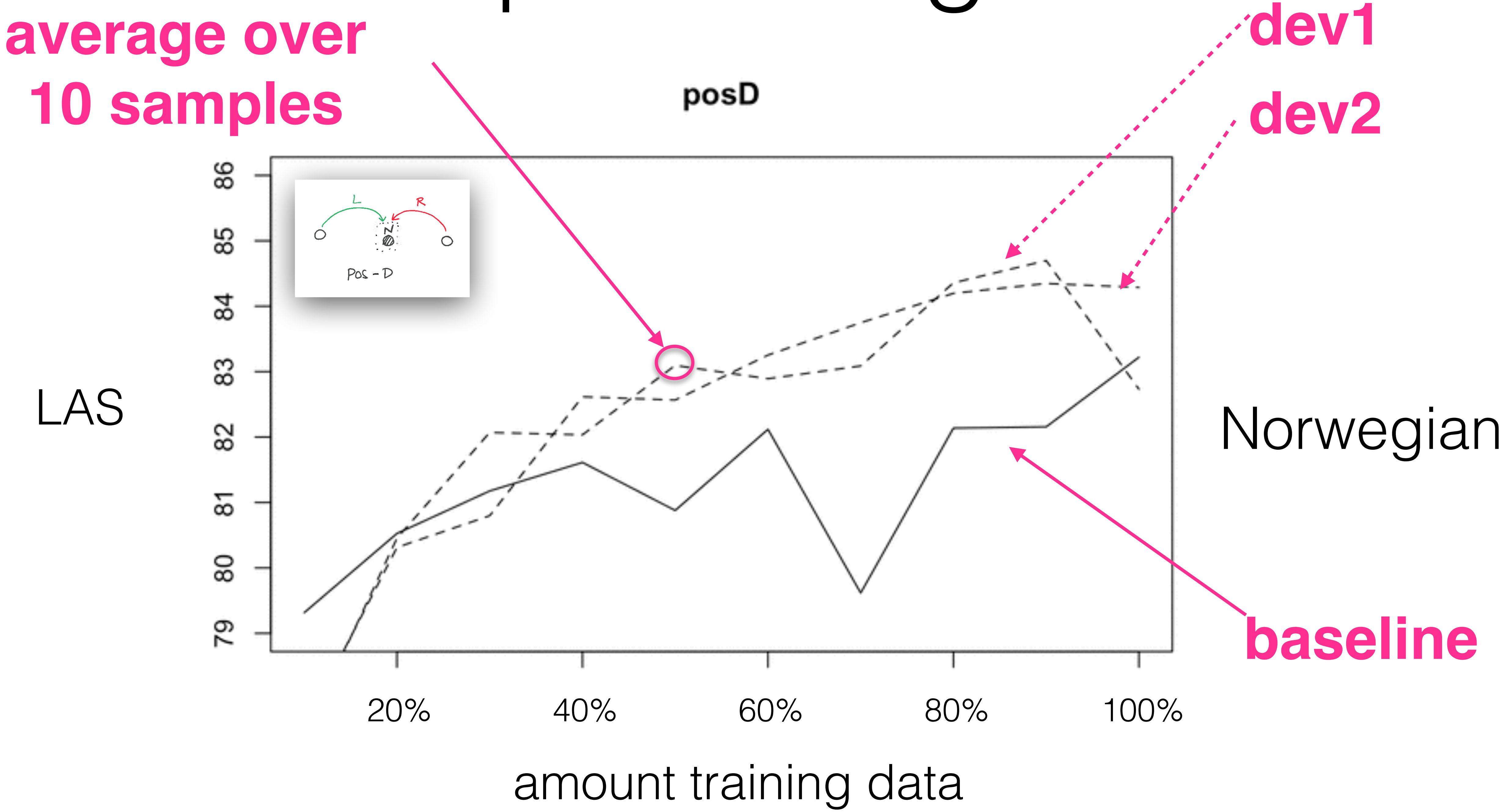
  - on down-stream tasks

- Estimate significance cut-off

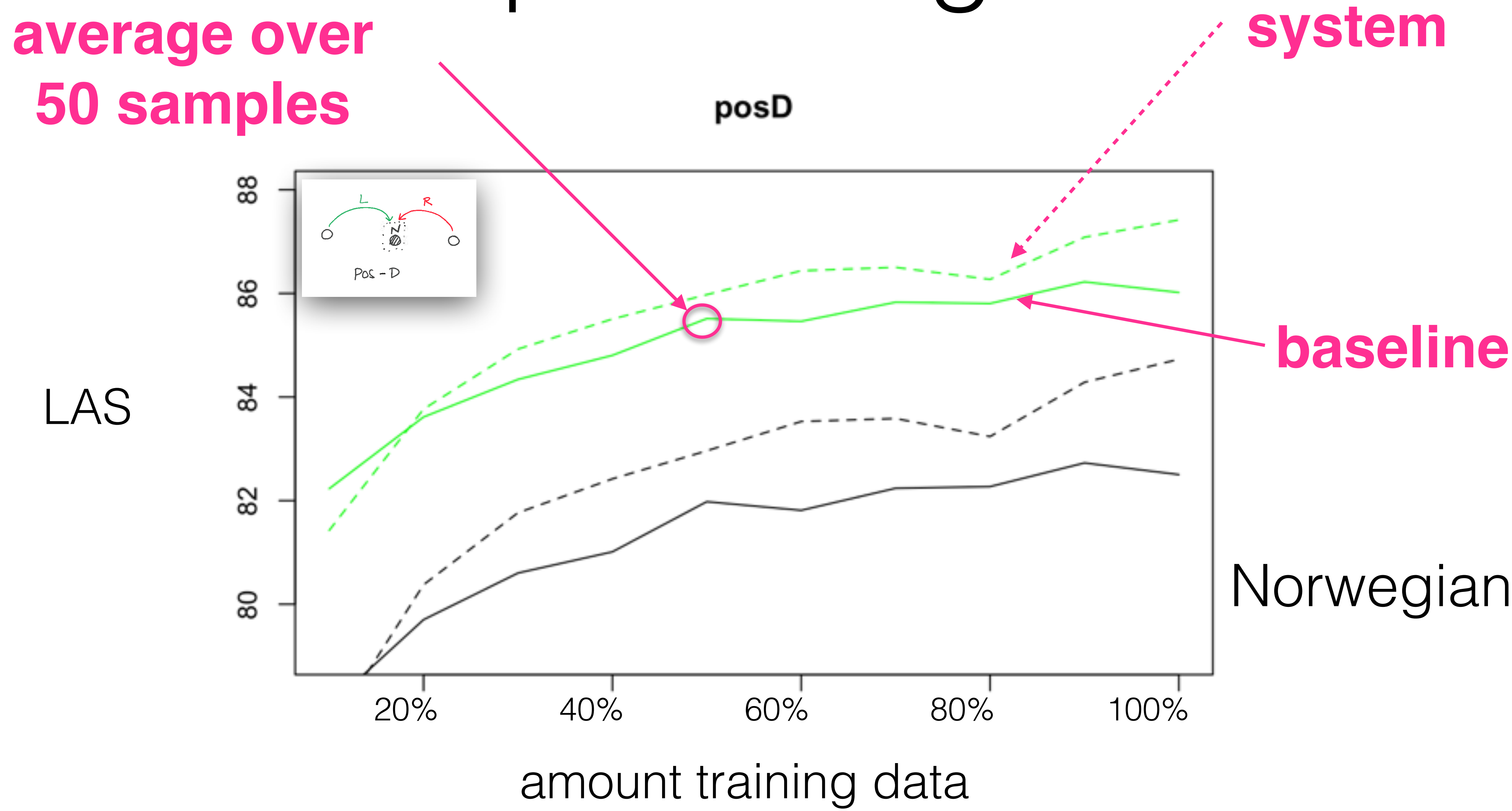
- Bootstrap-based evaluation



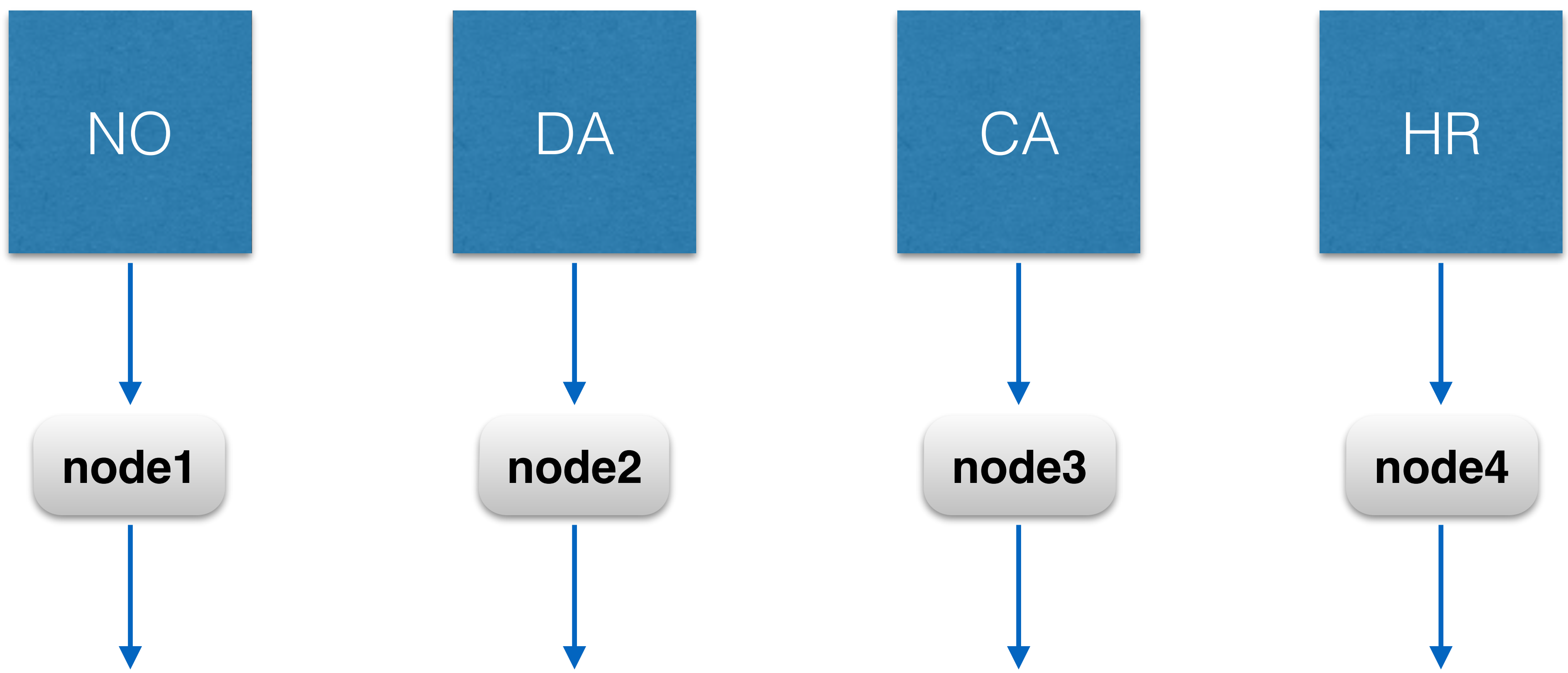
IAA-informed parsing: bootstrap learning curve



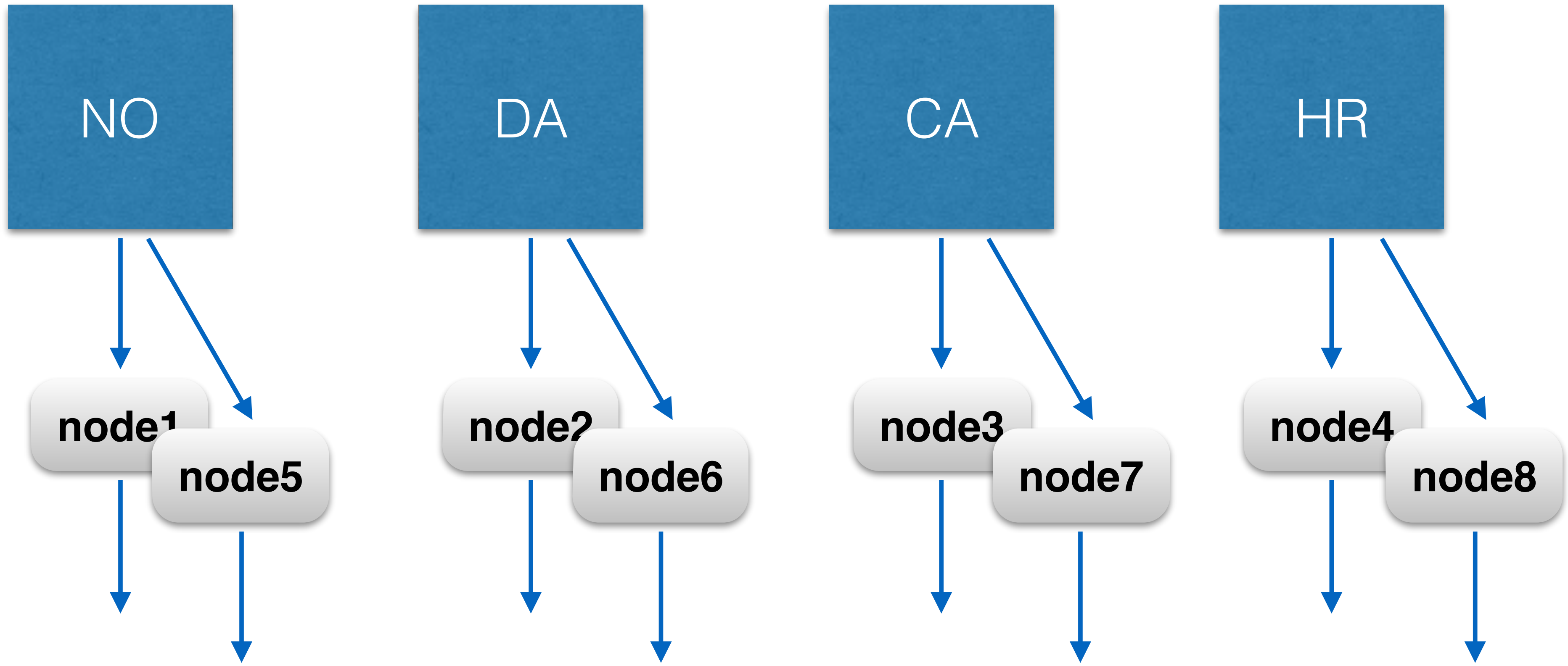
IAA-informed parsing: bootstrap learning curve



parallelization of NLP pipeline over 4 languages



parallelization of NLP pipeline over 4 languages



with 2 evaluation setups

sharing

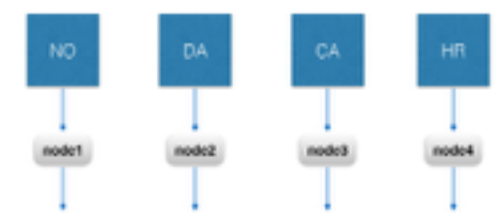
common data repository
for Nordic countries

Common Crawl

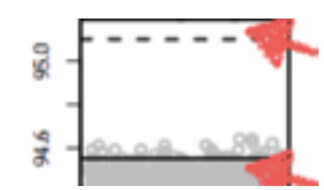
Summary: HPC for NLP

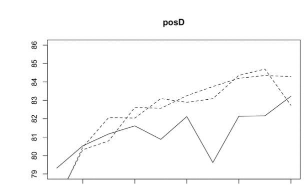
… besides parallel data processing and distributed training:

- **models:** parallelization over data sets, parameter search, negative results



- **evaluation:** significance cut-off, bootstrap samples





- **sharing:** common data repository

Thanks!