Language and Statistics II

Lecture 21: Bootstrapping

Noah Smith

So Far …

- We’ve talked mainly about building **models** from either annotated data or unannotated data.
- We’ve focused on classes of models that predict different kinds of **structure**.
- We’ve explored different ways to **estimate** those models.
- Today, we focus on mixing labeled and unlabeled data.

Word Sense Disambiguation

- Can a word sense disambiguation?
- Homographs
  - park the car *vs.* walk in the park
  - water the plant *vs.* work at the plant
  - the *x* and *y* axes *vs.* chopping down trees with axes
  - palm of my hand *vs.* palm tree
- Assume we know the set of senses for a word type. Can we pick the right one for ambiguous tokens in text?
- Note: the "output variable" ranges over a small, finite set. So machine learning people love WSD.

One Sense Per Discourse

p(more than one occurrence)

p(most frequent sense | more than one occurrence)

Word	Senses	Accuracy	Applicblty
plant	living/factory	99.8 %	72.8 %
tank	vehicle/contnr	99.6 %	50.5 %
poach	steal/boil	100.0 %	44.4 %
palm	tree/hand	99.8 %	38.5 %
axes	grid/tools	100.0 %	35.5 %
sake	benefit/drink	100.0 %	33.7 %
bass	fish/music	100.0 %	58.8 %
space	volume/outer	99.2 %	67.7 %
motion	legal/physical	99.9 %	49.8 %
crane	bird/machine	100.0 %	49.1 %
Average		**99.8 %**	**50.1 %**

One Sense Per Discourse

- This is a fancy way of saying that, within a discourse (e.g., document), ambiguous tokens of the same type tend to be **correlated**.

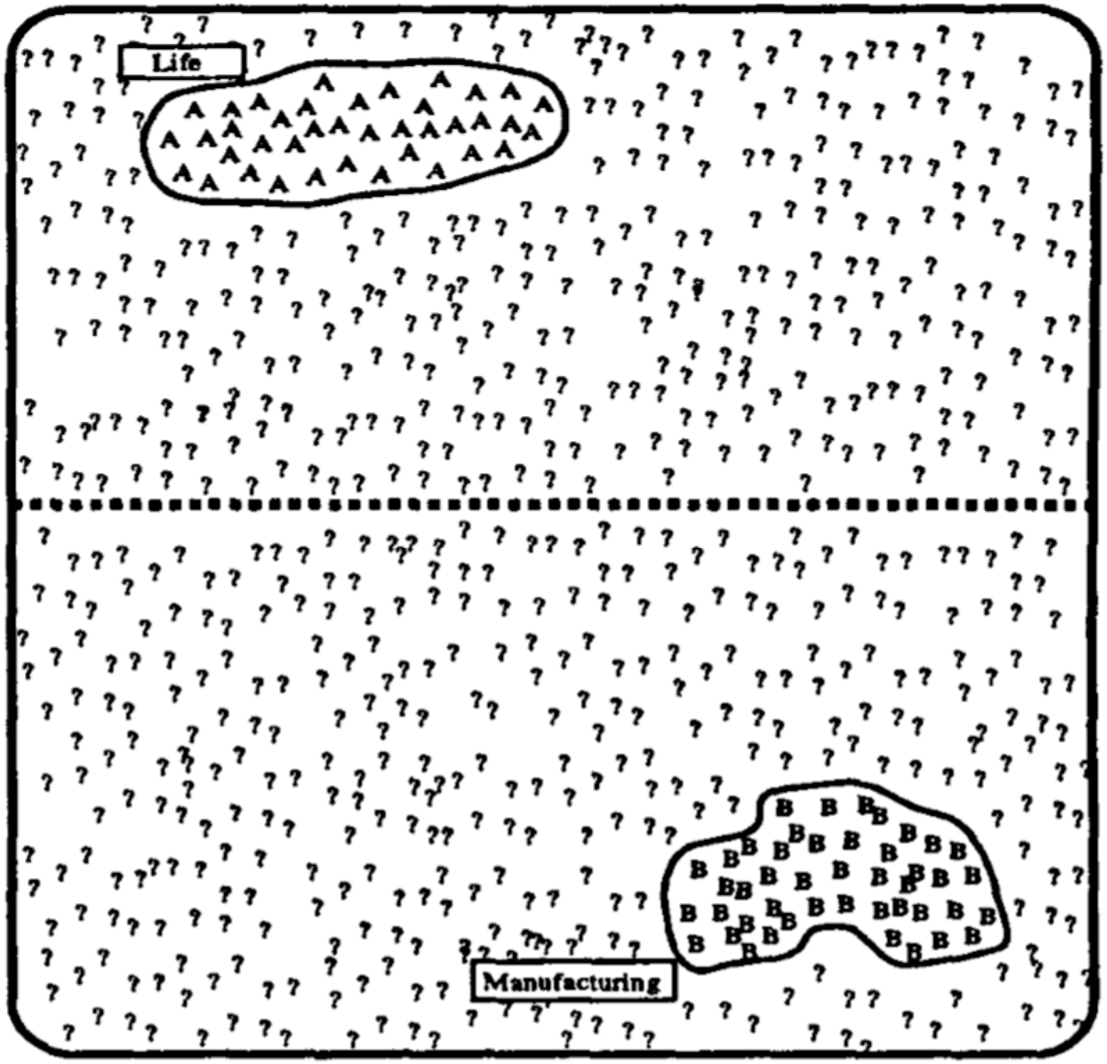
One Sense Per Collocation

- Certain features of the context are very strong predictors for one sense or another.
  - … power plant …
  - … palm of …
  - … the park …
- This is a fancy way of saying that (some) collocations are excellent features.

The Yarowsky Algorithm

- Given: ambiguous word type **w**, lots of text

1. Choose a few seed collocations for each sense and label data in those collocations.

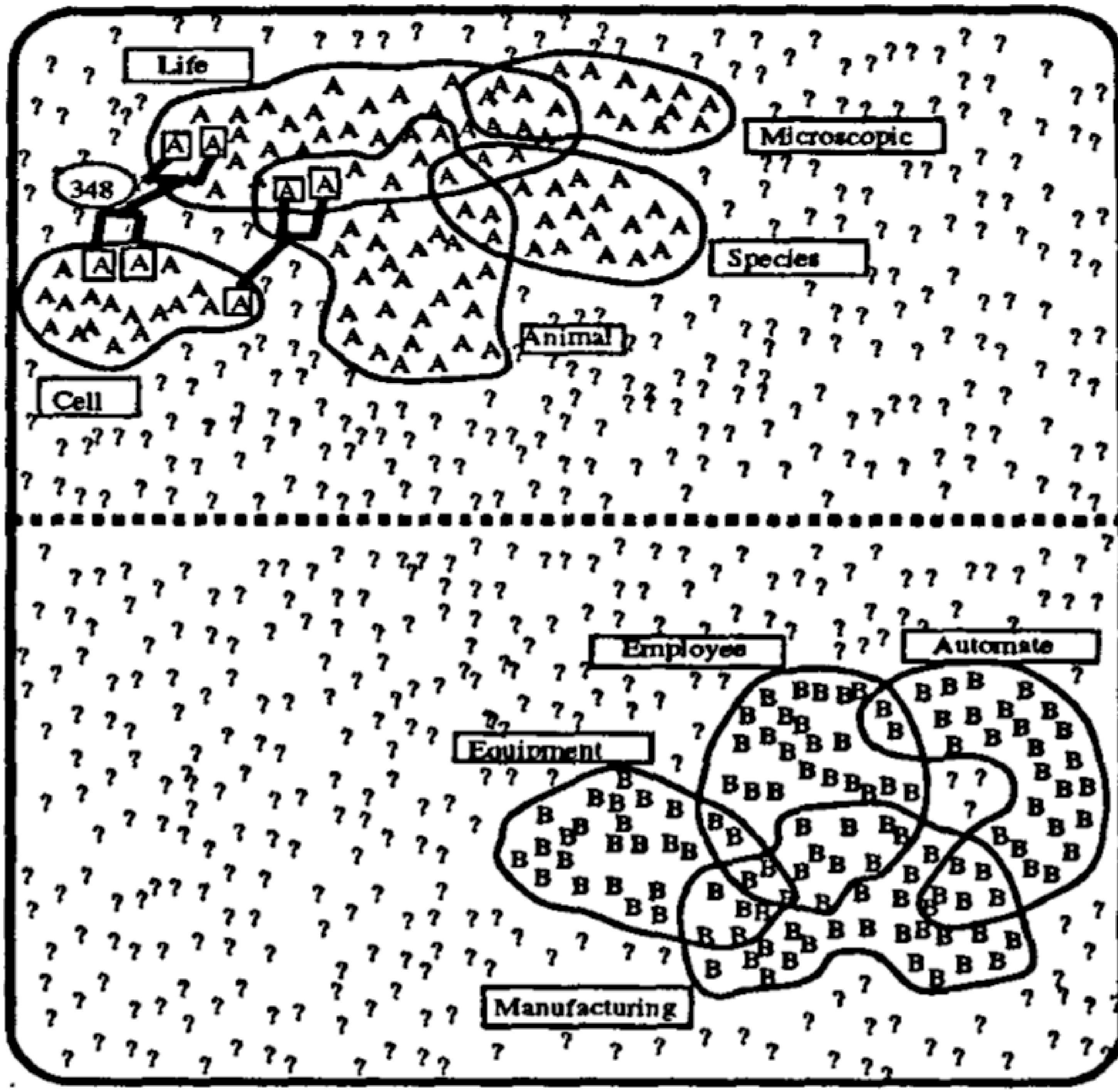


A = SENSE-A training example
B = SENSE-B training example
? = currently unclassified training example
$\boxed{\text{Life}}$ = Set of training examples containing the collocation "life".

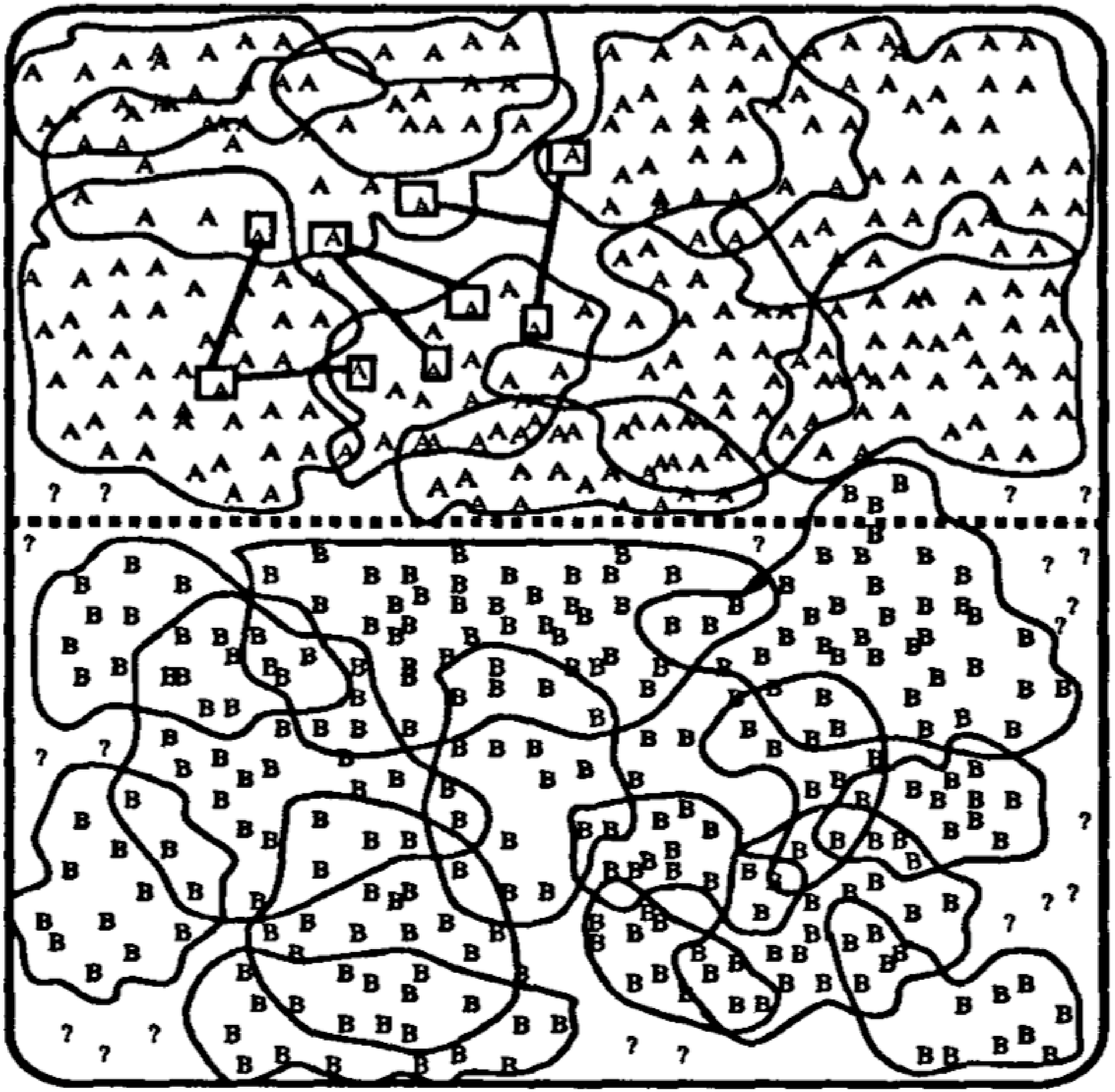
The Yarowsky Algorithm

- Given: ambiguous word type **w**, lots of text

1. Choose a few seed collocations for each sense and label data in those collocations.
2. Train a supervised classifier on the labeled examples. (Yarowsky used a decision list.)
3. Label all examples. Keep the labels about which the supervised classifier was highly confident (above threshold).
   - Optionally, exploit one-sense-per-discourse to “spread” a label throughout the discourse.
4. Go to 2.



Life
Microscopic
348
Species
Animal
Cell
Employee
Automate
Equipment
Manufacturing



Whence Seeds?

- Yarowsky suggests:
  - dictionary definitions
  - single defining collocate (e.g., from WordNet)
  - label extremely common collocations

- See Eisner & Karakos (2005) for more about seeds.

Experimental Results

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
			%		Seed Training Options			(7) + OSPD		
Word	Senses	Samp. Size	Major Sense	Supvsd Algrtm	Two Words	Dict. Defn.	Top Colls.	End only	Each Iter.	Schütze Algrthm
plant	living/factory	7538	53.1	97.7	97.1	97.3	97.6	98.3	98.6	92
space	volume/outer	5745	50.7	93.9	89.1	92.3	93.5	93.3	93.6	90
tank	vehicle/container	11420	58.2	97.1	94.2	94.6	95.8	96.1	96.5	95
motion	legal/physical	11968	57.5	98.0	93.5	97.4	97.4	97.8	97.9	92
bass	fish/music	1859	56.1	97.8	96.6	97.2	97.7	98.5	98.8	–
palm	tree/hand	1572	74.9	96.5	93.9	94.7	95.8	95.5	95.9	–
poach	steal/boil	585	84.6	97.1	96.6	97.2	97.7	98.4	98.5	–
axes	grid/tools	1344	71.8	95.5	94.0	94.3	94.7	96.8	97.0	–
duty	tax/obligation	1280	50.0	93.7	90.4	92.1	93.2	93.9	94.1	–
drug	medicine/narcotic	1380	50.0	93.0	90.4	91.4	92.6	93.3	93.9	–
sake	benefit/drink	407	82.8	96.3	59.6	95.8	96.1	96.1	97.5	–
crane	bird/machine	2145	78.0	96.6	92.3	93.6	94.2	95.4	95.5	–
AVG		3936	63.9	96.1	90.6	94.8	95.5	96.1	96.5	92.2

Several Ways to Think About This

- Like **Viterbi EM**, but new features induced on each iteration.
  - Yarowsky didn’t use a probability model in the conventional way; he used a decision list.
- Leveraging several assumptions about the data to help each other
  - One sense per collocation (inside the decision list)
  - One sense per discourse (finding new collocations)
- Meta-learner in which any supervised method can be nested!

Important Note

- Yarowsky's algorithm is not just for word sense! Similar algorithms have been applied to diverse problems:
  - Named entity recognition
  - Grammatical gender prediction
  - Morphology learning
  - Bilingual lexicon induction
  - Parsing

Cotraining (Blum and Mitchell, 1998)

- Rather difficult paper, but rather elegant idea.
- Input is x; suppose it can be broken into x_1 and x_2, disjoint “views” of x.
- Cotraining iteratively builds two classifiers (one on x_1 and one on x_2) and uses each to help improve the other.

Cotraining

- Given labeled examples L, unlabeled examples U

1. Train c_1 on x_1 from L, and train c_2 on x_2 from L. (B&M used Naïve Bayes.)
2. Label examples in U using c_1; add those it's most confident about for each class to L.
3. Ditto (c_2).
4. Go to 1.

WebKB-Course Data

- Data: CS department sites from four universities
- Task: Is a given page a course web page or not?
- X_1: bag of words in the page
- X_2: bag of words in *hyperlinks to the page*

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

What's Different?

- The "view" formulation.
  - Yarowsky has one classifier; B&M have two.
- Yarowsky allows relabeling of unlabeled examples; B&M do not.
- Yarowsky (1995) focused on particular properties of the data and exploited them. No general claims.
- B&M (1998) were seeking a general meta-learner that could leverage unlabeled examples; they actually gave PAC-style learnability results under an assumption that X_1 and X_2 were conditionally independent given Y.
- Unlike EM, neither of these methods maintains **posterior distributions** over the labels.

Nigam and Ghani (2000)

- Compare EM and cotraining, with the **same** model/featues. On the WebKB-Course dataset:

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	788	–0–	3.3%
Co-training	12	776	5.4%
EM	12	776	4.3%
Naive Bayes	12	–0–	13.0%

Nigam and Ghani (2000)

- Ceiling effects?
- Are the content/hyperlink views really independent? (Probably not.) Semi-synthetic experiment:

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	1006	–0–	3.9%
Co-training	6	1000	3.7%
EM	6	1000	8.9%
Naive Bayes	6	–0–	34.0%

- EM > Cotraining

Hybrids

- EM:

  ||: A softly labels data; A trains :||
- Co-EM:

  ||: A softly labels data; B trains;

  B softly labels data; A trains :||
- Co-training:

  ||: A, B label a few examples; A, B train :||
- Self-training:

  ||: A labels a few examples; A trains :||

Results (Synthetic Data)

cotraining self-training

Method	Uses Feature Split? Yes	No
Incremental	3.7%	5.8%
Iterative	3.3%	8.9%

co-EM EM

More Results

- If no natural feature split is available, can split features **randomly**.
- On synthetic data, that actually worked better than the smart split!
- On real data, best results came from self-training (!?!?)
  - Hard to draw any firm conclusions.
  - Possibly has to do with the supervised learner (why not use something more powerful than Naïve Bayes?).
  - Ng and Cardie (2003): more mixed results, but come out in favor of “single-view” algorithms.
  - Critical comment: go back to the objective function!

Abney (2004)

- “Understanding the Yarowsky Algorithm”
- Entirely under-appreciated paper!
- Demonstrates that certain variants of the Yarowsky algorithm are actually optimizing likelihood. Others are optimizing a bound on likelihood.
- Likelihood under what model?

Understanding the Abney Understanding of the Yarowsky Algorithm

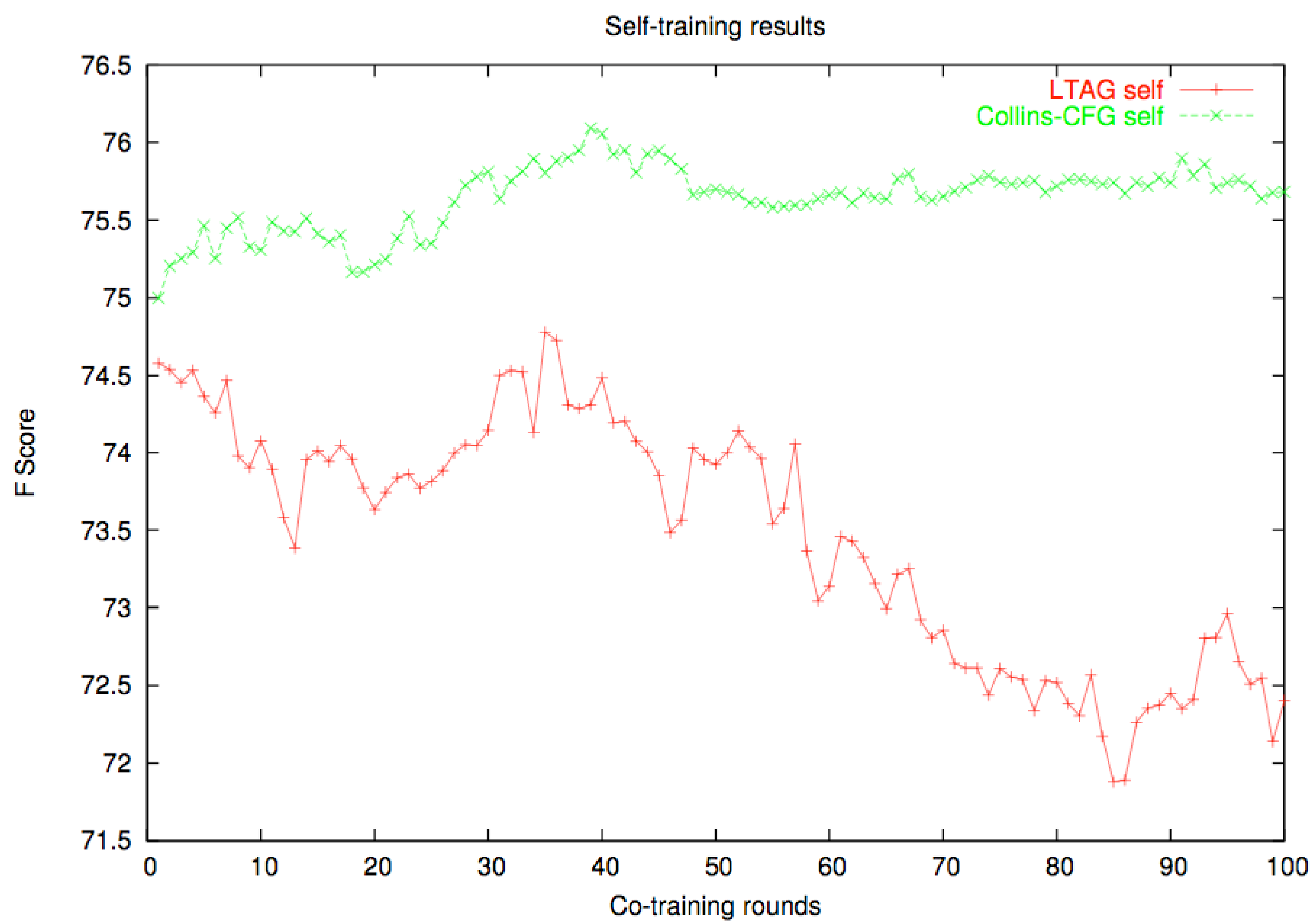
- Modifications:
  - Once an originally-unlabeled example is labeled, it stays labeled.
  - Fix threshold at 1/(# classes).
- Assumption: base learner *improves* **KL divergence** between empirical distribution and the base model.
  - Either on labeled examples only,
  - or overall (assuming unlabeled examples have uniform empirical)
- Yarowsky's base learner doesn't do this; Abney gives variants that do.
  - The "DL-EM" base learners he describes essentially amount to a single step of the EM algorithm.
- The proofs are involved; the insight (I believe) is that the algorithm starts to look more like (Viterbi) EM with some labels fixed so they can't change.

Cotraining for Parsing?

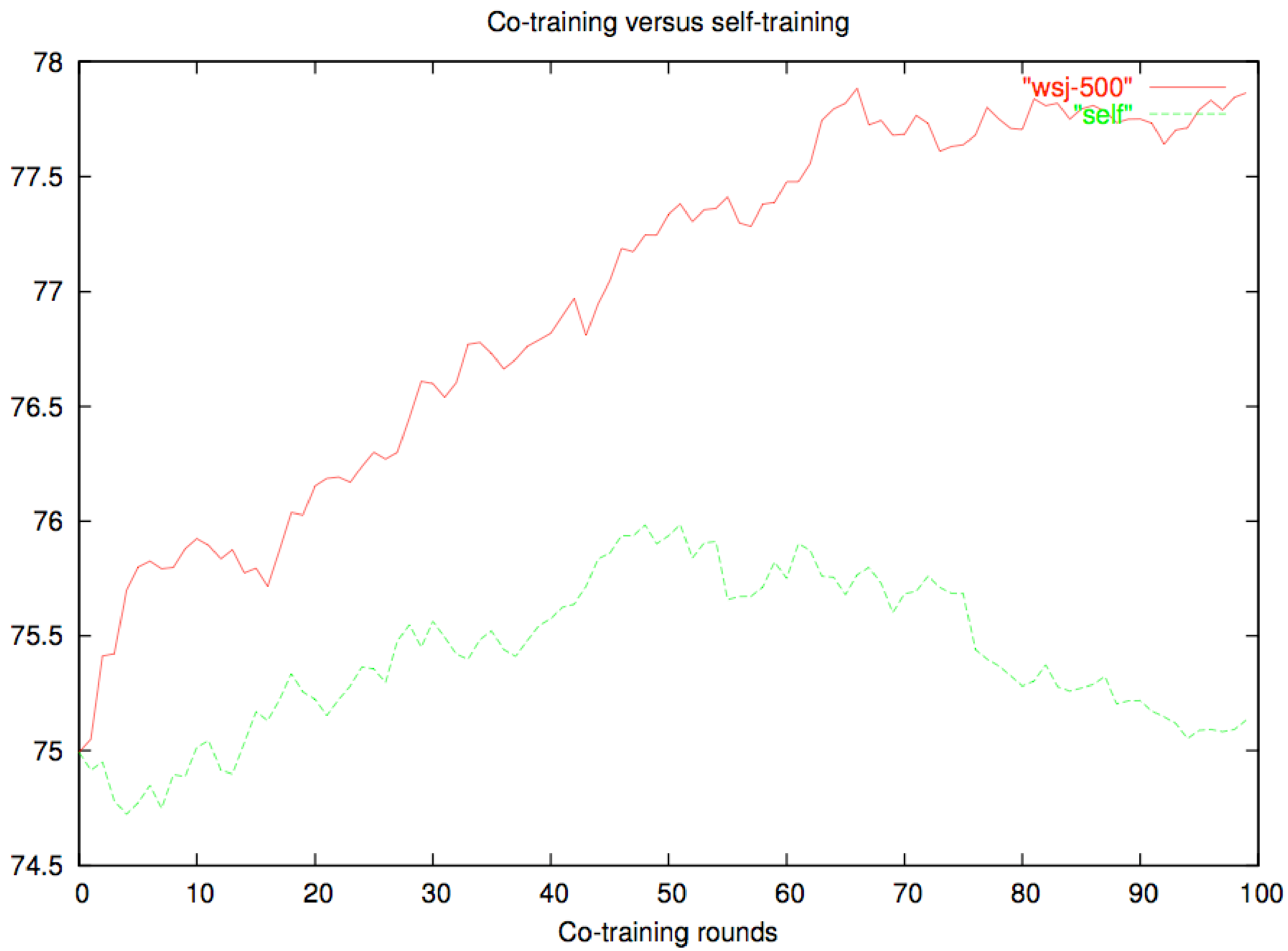
- Steedman et al. (2003) cotrained two parsers.

Collins-CFG	LTAG
Bi-lexical dependencies are between lexicalized nonterminals	Bi-lexical dependencies are between elementary trees
Can produce novel elementary trees for the LTAG parser	Can produce novel bi-lexical dependencies for Collins-CFG
When using small amounts of seed data, abstains less often than LTAG	When using small amounts of seed data, abstains more often than Collins-CFG

Parser Self-training



Parser Cotraining



Steedman et al., 2003

- Also showed cross-domain improvement (WSJ and Brown corpus).
- If you start with “enough” labeled data, cotraining doesn’t help.
  – Similar to many other results: Merialdo (1994), Elworthy (1994), Smith (2006), …

Semisupervised Learning: Hot

- Adaptation to new domains
  - Or languages! Hwa et al., 2002; Wicentowski et al., 2001; Smith and Smith, 2004, …
- Ando and Zhang (2005): use multiple tasks to leverage unlabeled data
- Lessen the cost of annotation projects (annotate fewer examples)
- Interesting theoretical topic (many papers lately)
- So much unlabeled data, how could we not want to learn from it!

Two Important Lessons

- There usually is no unqualified “best” method. All kinds of things affect this. More subtle questions than, “does A beat B”:
  - What conditions lead to better performance for A vs. B?
  - What kinds of errors is A more susceptible to than B?
- Nifty ideas can often be shown (sometimes years later) to have solid mathematical underpinnings.