

Bootstrapping Statistical Parsers from Small Datasets

Anoop Sarkar
Department of Computing Science
Simon Fraser University

`anoop@cs.sfu.ca`
`http://www.cs.sfu.ca/~anoop`

Overview

- Task: find the most likely parse for natural language sentences
- Approach: rank alternative parses with statistical methods trained on data annotated by experts (labelled data)
- Focus of this talk:
 1. Machine learning by combining different methods in parsing: [PCFG](#) and [Tree-adjoining grammar](#)
 2. Weakly supervised learning: combine labelled data with unlabelled data to improve performance in parsing using [co-training](#)

A Key Problem in Processing Language: Ambiguity: (Church and Patil 1982; Collins 1999)

- Part of Speech ambiguity

saw → noun

saw → verb

- Structural ambiguity: Prepositional Phrases

I saw (the man) with the telescope

I saw (the man with the telescope)

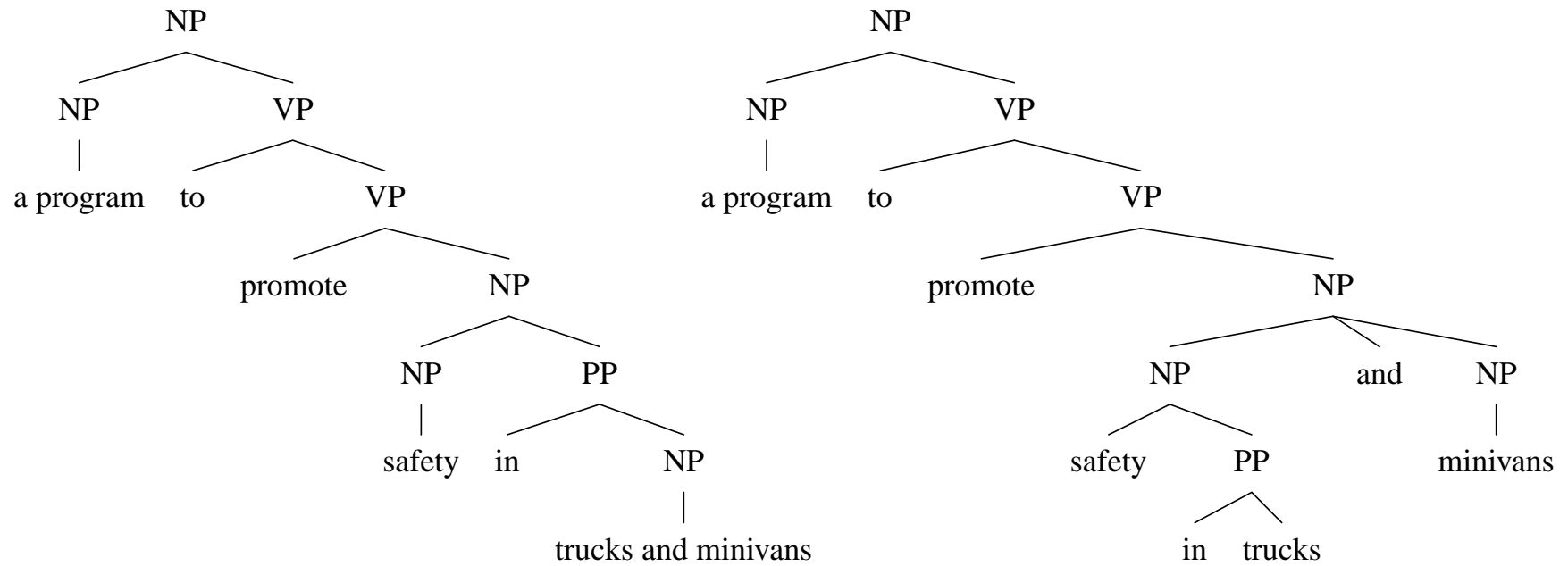
- Structural ambiguity: Coordination

a program to promote safety in ((trucks) and (minivans))

a program to promote ((safety in trucks) and (minivans))

((a program to promote safety in trucks) and (minivans))

Ambiguity ← attachment choice in alternative parses



Parsing as a machine learning problem

- S = a sentence
 T = a parse tree
A statistical parsing model defines $P(T | S)$
- Find best parse: $\arg \max_T P(T | S)$
- $P(T | S) = \frac{P(T, S)}{P(S)} = P(T, S)$
- Best parse: $\arg \max_T P(T, S)$
- e.g. for PCFGs: $P(T, S) = \prod_{i=1 \dots n} P(\text{RHS}_i | \text{LHS}_i)$

Parsing as a machine learning problem

- Training data: the Penn WSJ Treebank (Marcus et al. 1993)
- Learn probabilistic grammar from training data
- Evaluate accuracy on test data
- A standard evaluation:
Train on 40,000 sentences
Test on 2,300 sentences
- The simplest technique: PCFGs perform badly
Reason: not sensitive to the words

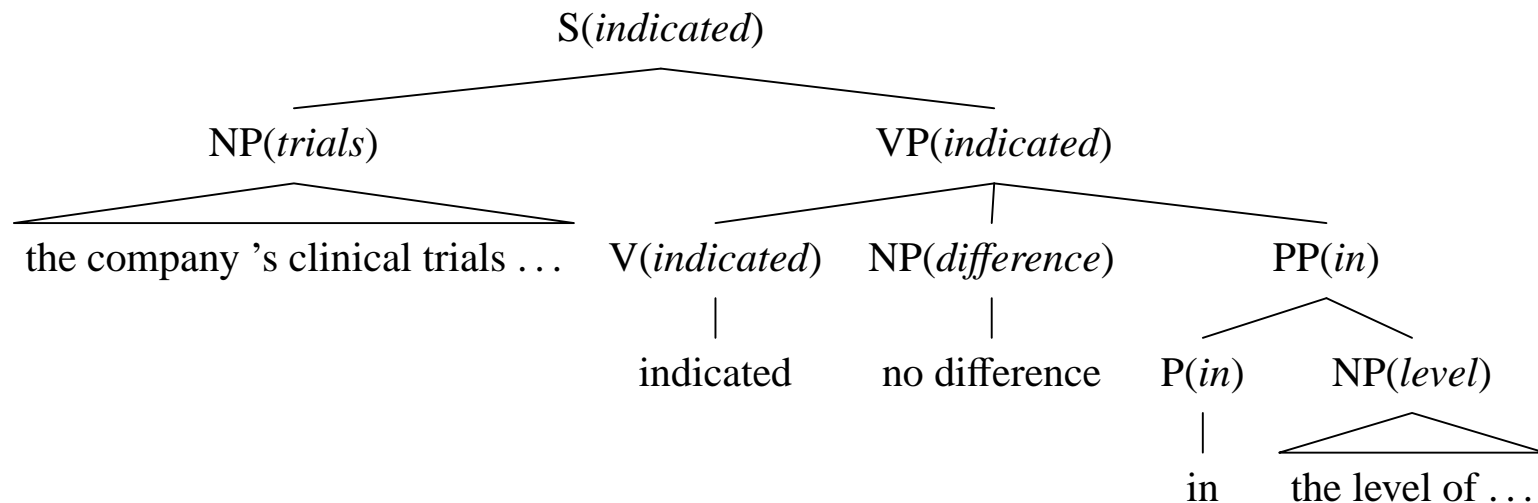
Machine Learning for ambiguity resolution: prepositional phrases

- What is right analysis for:
Calvin saw the car on the hill with the telescope
- Compare with:
Calvin bought the car with anti-lock brakes and
Calvin bought the car with a loan
- *(bought, with, brakes)* and *(bought, with, loan)* are useful features to solve this apparently AI-complete problem

Method	Accuracy
Always noun attachment	59.0
Most likely for each preposition	72.2
Average Human (4 head words only)	88.2
Average Human (whole sentence)	93.2
Lexicalized Model (Collins and Brooks 1995)	84.5
Lexicalized Model + Wordnet (Stetina and Nagao 1998)	88.0

Statistical Parsing

the company 's clinical trials of both its animal and human-based insulins indicated no difference in the level of hypoglycemia between users of either product



Use a probabilistic *lexicalized* grammar from the Penn WSJ Treebank for parsing ...

Bilexical CFG (Collins-CFG): dependencies between pairs of words

- Full context-free rule:

$VP(indicated) \rightarrow V\text{-hd}(indicated) NP(difference) PP(in)$

- Each rule is generated in three steps (Collins 1999):

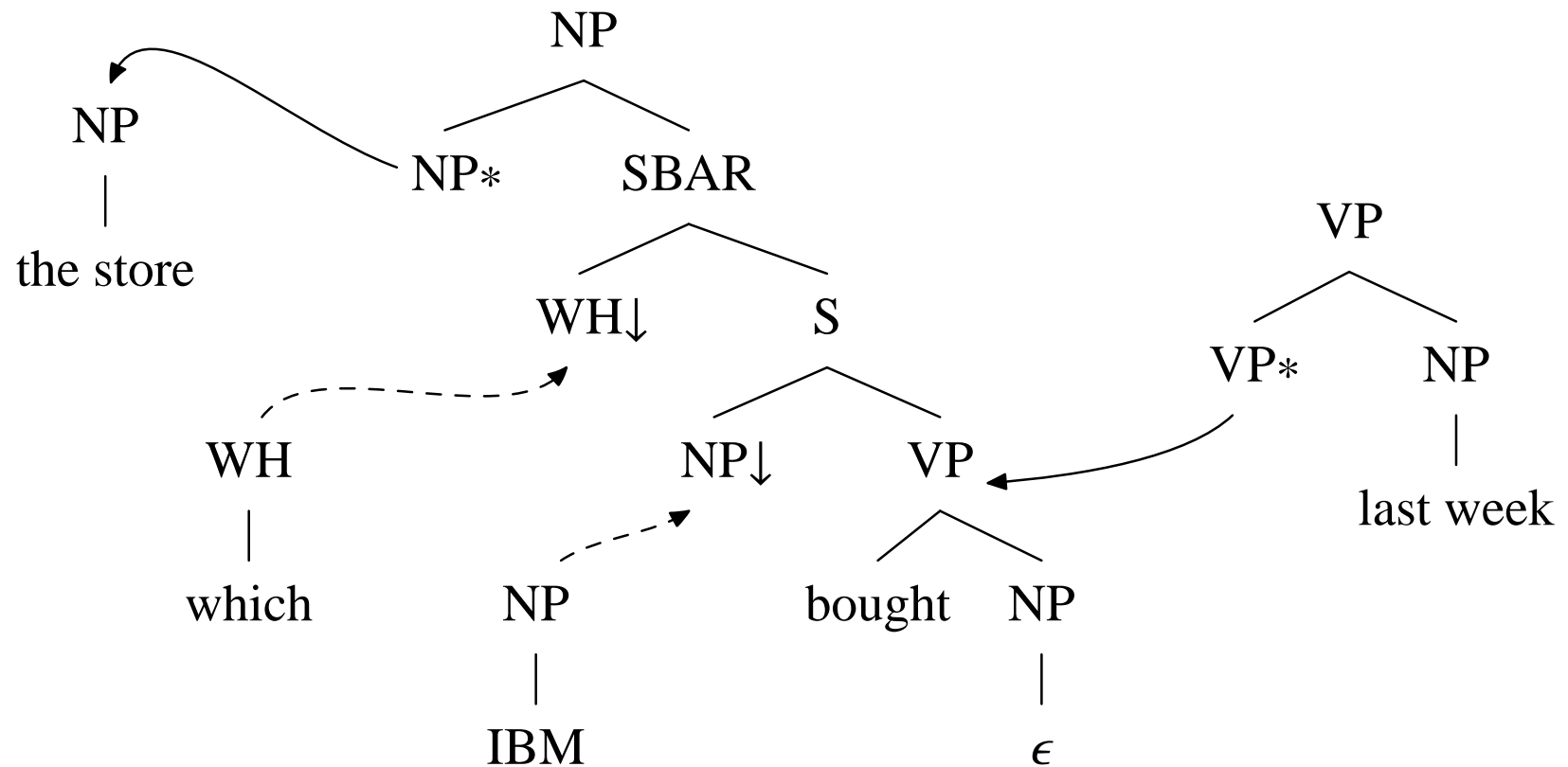
1. Generate head daughter of LHS: $VP(indicated) \rightarrow V\text{-hd}(indicated)$

2. Generate non-terminals to *left* of head daughter: $STOP \dots$
 $V\text{-hd}(indicated)$

3. Generate non-terminals to *right* of head daughter:

- V-hd(*indicated*) ... NP(*difference*)
- V-hd(*indicated*) ... PP(*in*)
- V-hd(*indicated*) ... STOP

Lexicalized Tree Adjoining Grammars (LTAG):
Different Modeling of Bilexical Dependencies



Performance of supervised statistical parsers

System	$\leq 40wds$ LP	$\leq 40wds$ LR	$\leq 100wds$ LP	$\leq 100wds$ LR
PCFG (Collins 99)	88.5	88.7	88.1	88.3
LTAG (Sarkar 01)	88.63	88.59	87.72	87.66
LTAG (Chiang 00)	87.7	87.7	86.9	87.0
PCFG (Charniak 99)	90.1	90.1	89.6	89.5
Re-ranking (Collins 00)	90.1	90.4	89.6	89.9

- Labelled Precision = $\frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in proposed parse}}$
- Labelled Recall = $\frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in treebank parse}}$

Bootstrapping

- Current state-of-the-art in parsing on the Penn WSJ Treebank dataset is approx 90% accuracy
- However this accuracy is obtained with **1M words** of human annotated data (40K sentences)
- Exploring methods that can exploit unlabelled data is an important goal:
 - What about different languages? The Penn Treebank took several years with many linguistic experts and millions of dollars to produce. Unlikely to happen for all other languages of interest.

- What about different genres? Porting a parser trained on newspaper text and using it on fiction is a challenge.
 - Combining labelled and unlabelled data is an interesting challenge for machine learning.
- In this talk, we will consider *bootstrapping* using unlabelled data.
- Bootstrapping refers to a problem setting in which one is given a small set of labelled data and a large set of unlabelled data, and the task is to extract new labelled instances from the unlabelled data.
- The noise introduced by the new automatically labelled instances has to be offset by the utility of training on those instances.

Multiple Learners and the Bootstrapping problem

- With a single learner, the simplest method of bootstrapping is called *self-training*.
- The high precision output of a classifier can be treated as new labelled instances (Yarowsky, 1995).
- With multiple learners, we can exploit the fact that they might:
 - Pay attention to different features in the labelled data.
 - Be confident about different examples in the unlabelled data.
 - Combine multiple learners using the *co-training* algorithm.

Co-training

- Pick two “views” of a classification problem.
- Build separate models for each of these “views” and train each model on a small set of labelled data.
- Sample an unlabelled data set and to find examples that each model independently labels with high confidence.
- Pick confidently labelled examples and add to labelled data. Iterate.
- Each model labels examples for the other in each iteration.

An Example: (Blum and Mitchell 1998)

- Task: Build a classifier that categorizes web pages into two classes, $+$: *is a course web page*, $-$: *is not a course web page*
- Usual model: build a Naive Bayes model:

$$P[C = c_k \mid X = \mathbf{x}] = \frac{P(c_k) \times P(\mathbf{x} \mid c_k)}{P(\mathbf{x})}$$
$$P(\mathbf{x} \mid c_k) = \prod_{x_j \in \mathbf{x}} P(x_j \mid c_k)$$

- Each labelled example has two views:

\mathbf{x}_1 Text in hyperlink: ``CSE 120, Fall semester``

\mathbf{x}_2 Text in web page: `<html>...`Assignment #1`...</html>`

- Documents in the unlabelled data where $C = c_k$ is predicted with high confidence by classifier trained on view \mathbf{x}_1 can be used as new training data for view \mathbf{x}_2 and vice versa
- Each view can be used to create new labelled data for the other view.
- Combining labelled and unlabelled data in this manner outperforms using only the labelled data.

Theory behind co-training: (Abney, 2002)

- For each instance x , we have two views $X_1(x) = x_1, X_2(x) = x_2$. x_1, x_2 satisfy *view independence* if:

$$Pr[X_1 = x_1 \mid X_2 = x_2, Y = y] = Pr[X_1 = x_1 \mid Y = y]$$

$$Pr[X_2 = x_2 \mid X_1 = x_1, Y = y] = Pr[X_2 = x_2 \mid Y = y]$$

- If $\mathcal{H}_1, \mathcal{H}_2$ are rules that use only X_1, X_2 respectively, then *rule independence* is:

$$Pr[F = u \mid G = v, Y = y] = Pr[F = u \mid Y = y]$$

where $F \in \mathcal{H}_1$ and $G \in \mathcal{H}_2$ (note that view independence implies rule independence)

Theory behind co-training: (Abney, 2002)

- Deviation from conditional independence:

$$d_y = \frac{1}{2} \sum_{u,v} | Pr[G = v \mid Y = y, F = u] - Pr[G = v \mid Y = y] |$$

- For all $F \in \mathcal{H}_1, G \in \mathcal{H}_2$ such that

$$d_y \leq p_2 \frac{q_1 - p_1}{2p_1q_1}$$

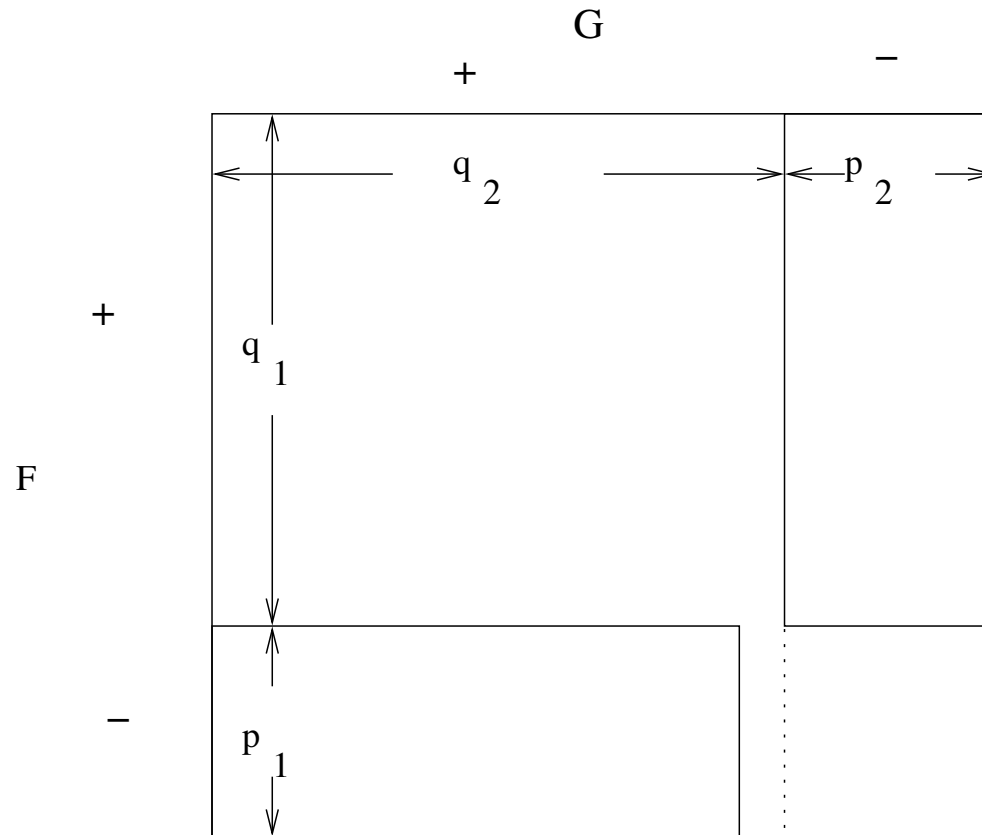
and $\min_u Pr[F = u] > Pr[F \neq G]$ then

$$Pr[F \neq Y] \leq Pr[F \neq G]$$

$$Pr[\bar{F} \neq Y] \leq Pr[F \neq G]$$

we can choose between F and \bar{F} using seed labelled data

Theory behind co-training: $Pr[F \neq Y] \leq Pr[F \neq G]$



Positive Correlation, $Y = +$

Theory behind co-training

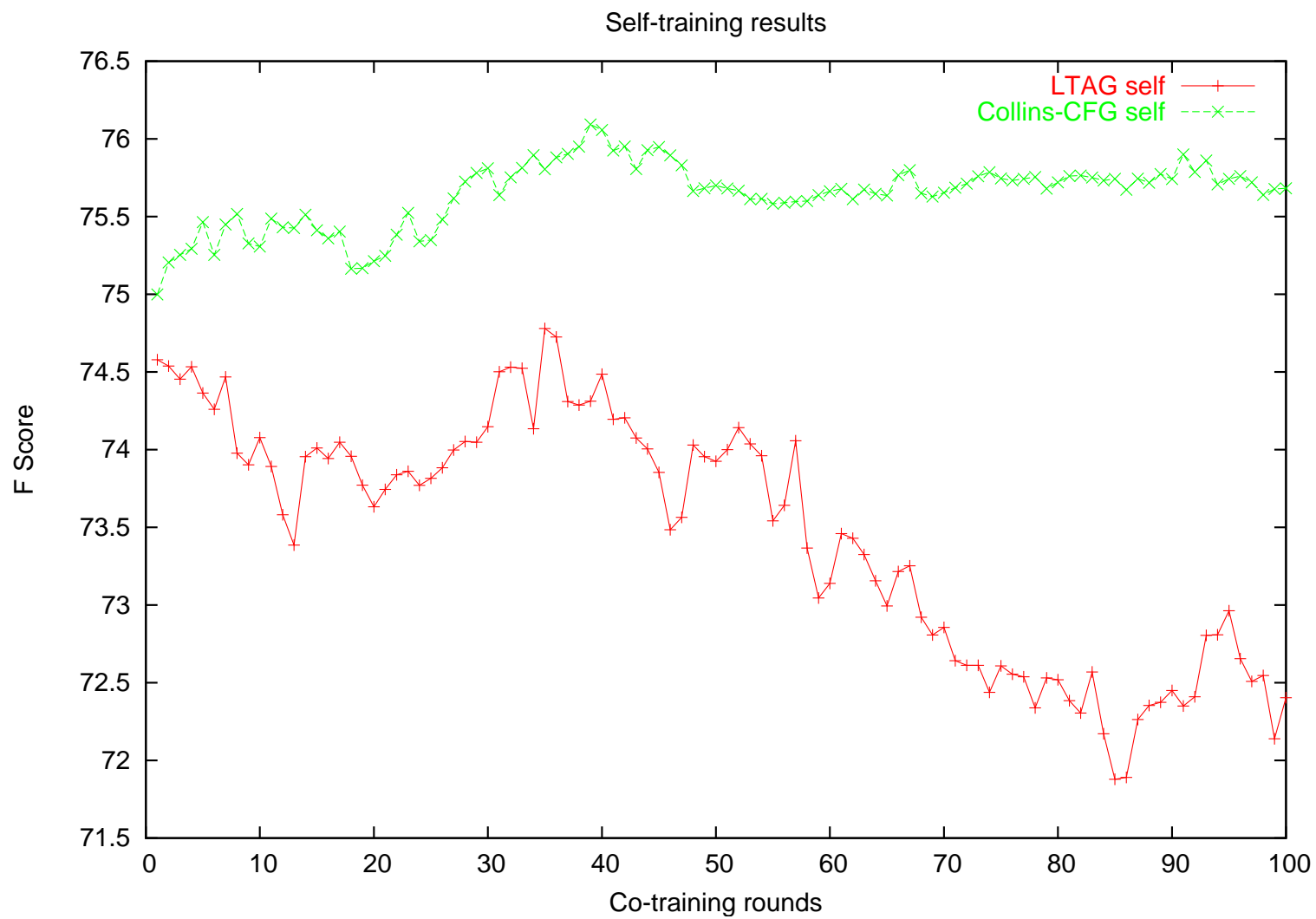
- (Blum and Mitchell, 1998) prove that, when the two views are *conditionally independent* given the label, and each view is sufficient for learning the task, co-training can improve an initial weak learner using unlabelled data.
- (Dasgupta et al, 2002) show that maximising the agreement over the unlabelled data between two learners leads to few generalisation errors (same independence assumption).
- (Abney, 2002) argues that the independence assumption is extremely restrictive and typically violated in the data. He proposes a weaker independence assumption and a greedy algorithm that maximises agreement on unlabelled data.

Co-training for statistical parsing In order to conduct co-training experiments between statistical parsers, it was necessary to choose two parsers that generate comparable output but use different statistical models.

1. The Collins lexicalized PCFG parser (Collins, 1999), model 2. Some code for (re)training this parser was added to make the co-training experiments possible. We refer to this parser as **Collins-CFG**.
2. The Lexicalized Tree Adjoining Grammar (LTAG) parser of (Sarkar, 2001), which we refer to as the **LTAG** parser.

Summary of the Different Views

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
Using small amounts of seed data, abstains less often than LTAG	Using small amounts of seed data, abstains more often than Collins-CFG



The pseudo-code for the co-training algorithm

A and B are two different parsers.

M_A^i and M_B^i are models of A and B at step i .

U is a large pool of unlabelled sentences.

U^i is a small cache holding subset of U at step i .

L is the manually labelled seed data.

L_A^i and L_B^i are the labelled training examples
for A and B at step i .

Initialize:

$$L_A^0 \leftarrow L_B^0 \leftarrow L.$$

$$M_A^0 \leftarrow \text{Train}(A, L_A^0)$$

$$M_B^0 \leftarrow \text{Train}(B, L_B^0)$$

Loop:

$U^i \leftarrow$ Add unlabelled sentences from U .

M_A^i and M_B^i parse the sentences in U^i
and assign scores to them according to
their scoring functions f_A and f_B .

Select new parses $\{P_A\}$ and $\{P_B\}$
according to some selection method S ,
which uses the scores from f_A and f_B .

L_A^{i+1} is L_A^i augmented with $\{P_B\}$

L_B^{i+1} is L_B^i augmented with $\{P_A\}$

$M_A^{i+1} \leftarrow \text{Train}(A, L_A^{i+1})$

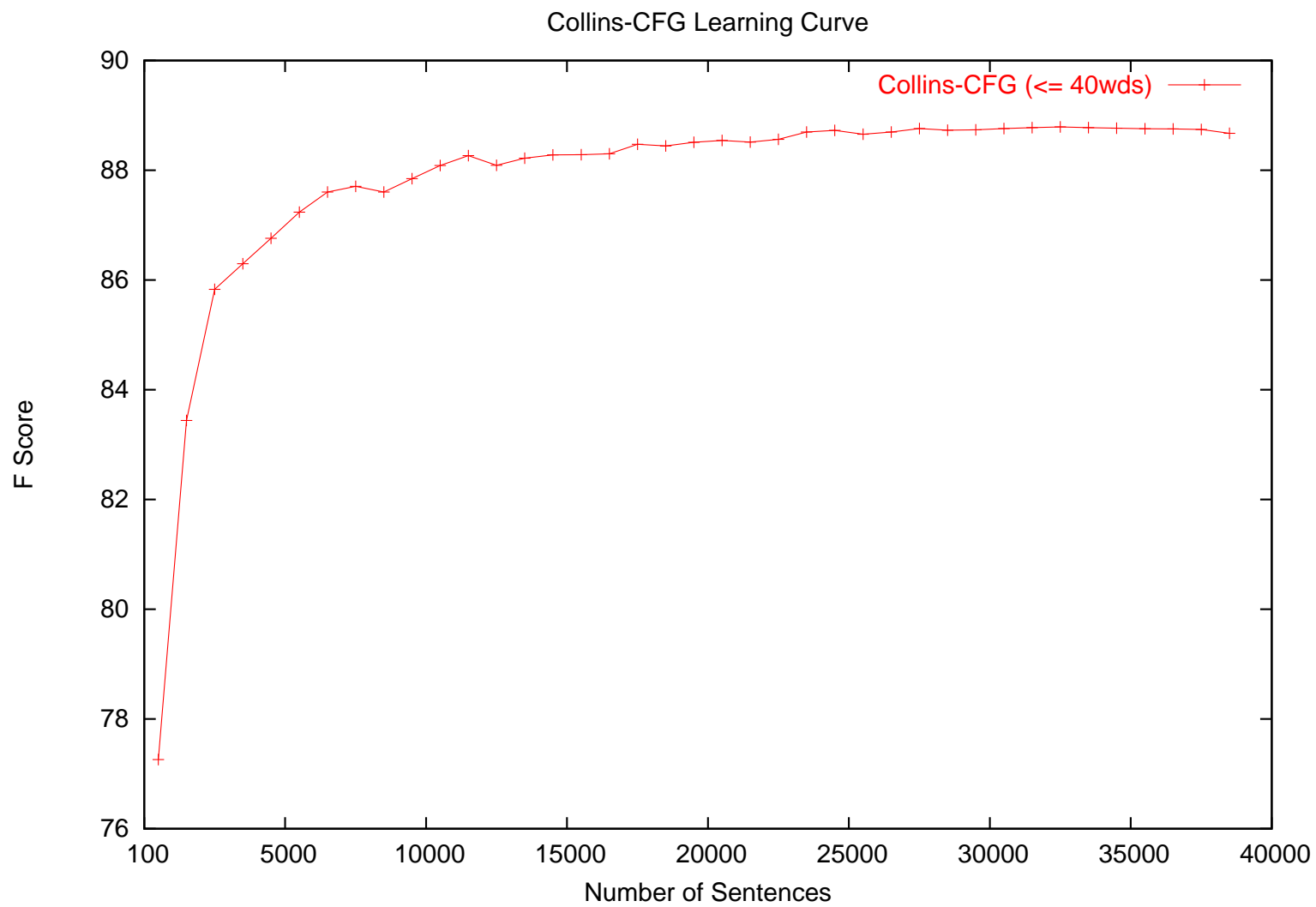
$M_B^{i+1} \leftarrow \text{Train}(B, L_B^{i+1})$

Experiments

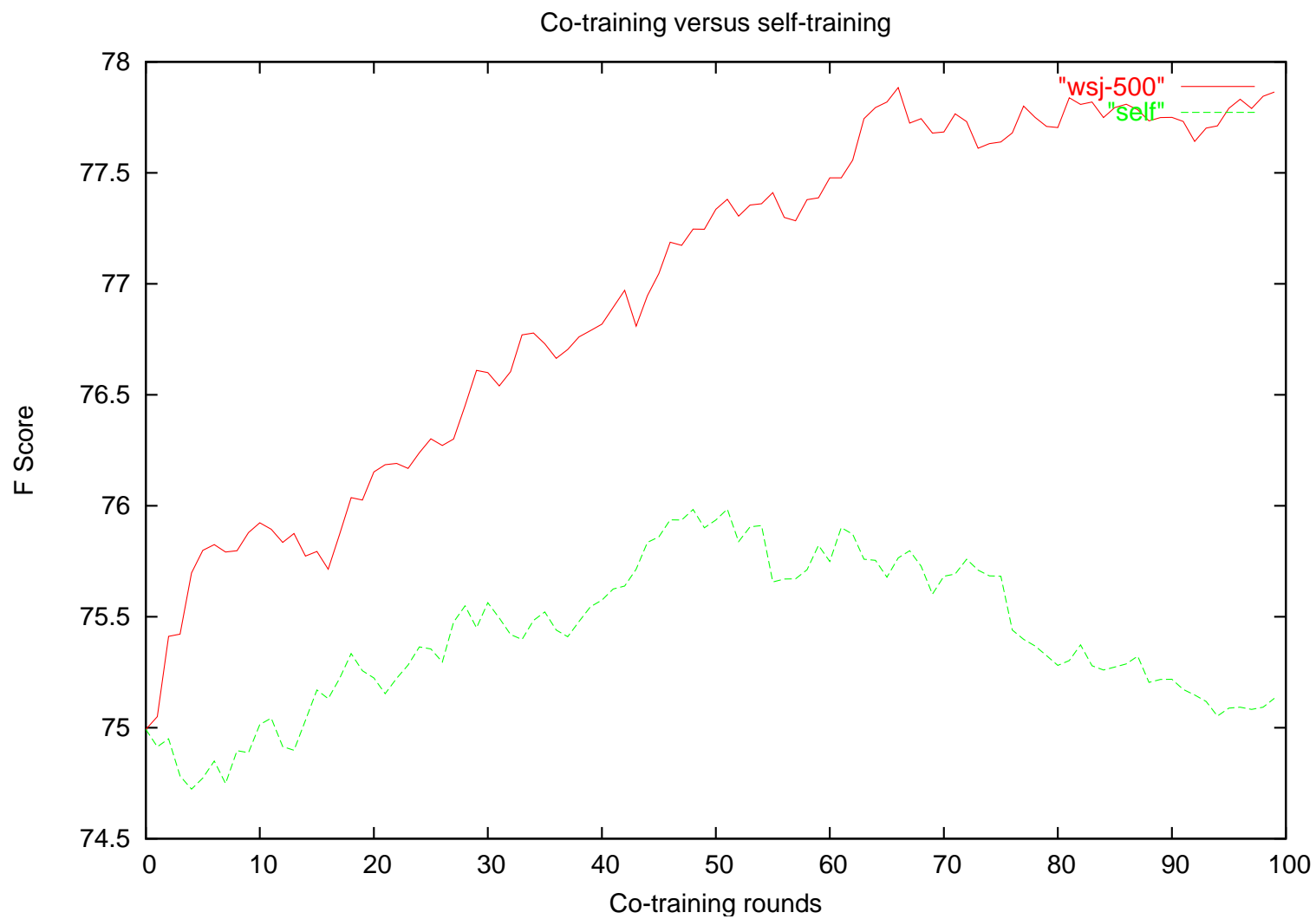
- Use co-training to boost performance, when faced with small seed data
→ Use small subsets of WSJ labelled data as seed data
- Use co-training to port parsers to new genres
→ Use Brown corpus as seed data, co-train and test on WSJ
- Use a large set of labelled data and use unlabelled data to improve parsing performance
→ Use Penn Treebank (40K sents) as seed data

Experiments on Small Labelled Seed Data

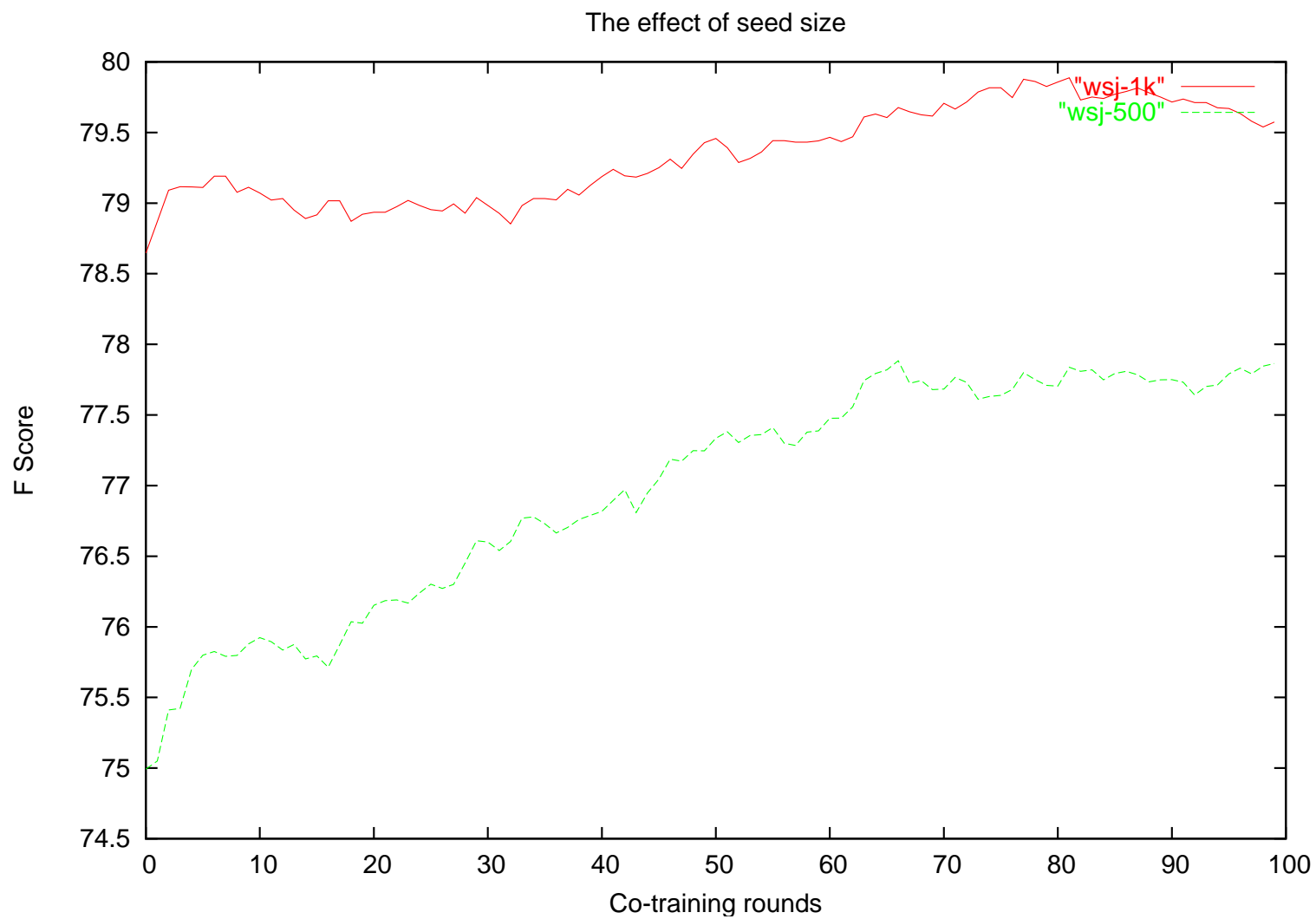
- Motivating the size of the initial seed data set
- We plotted learning curves, tracking parser accuracy while varying the amount of labelled data
- Find the “elbow” in the curve where the payoff will occur
- This was done for both the Collins-CFG and the LTAG parser
- The learning curve shows that the maximum payoff from co-training is likely to occur between 500 and 1,000 sentences.



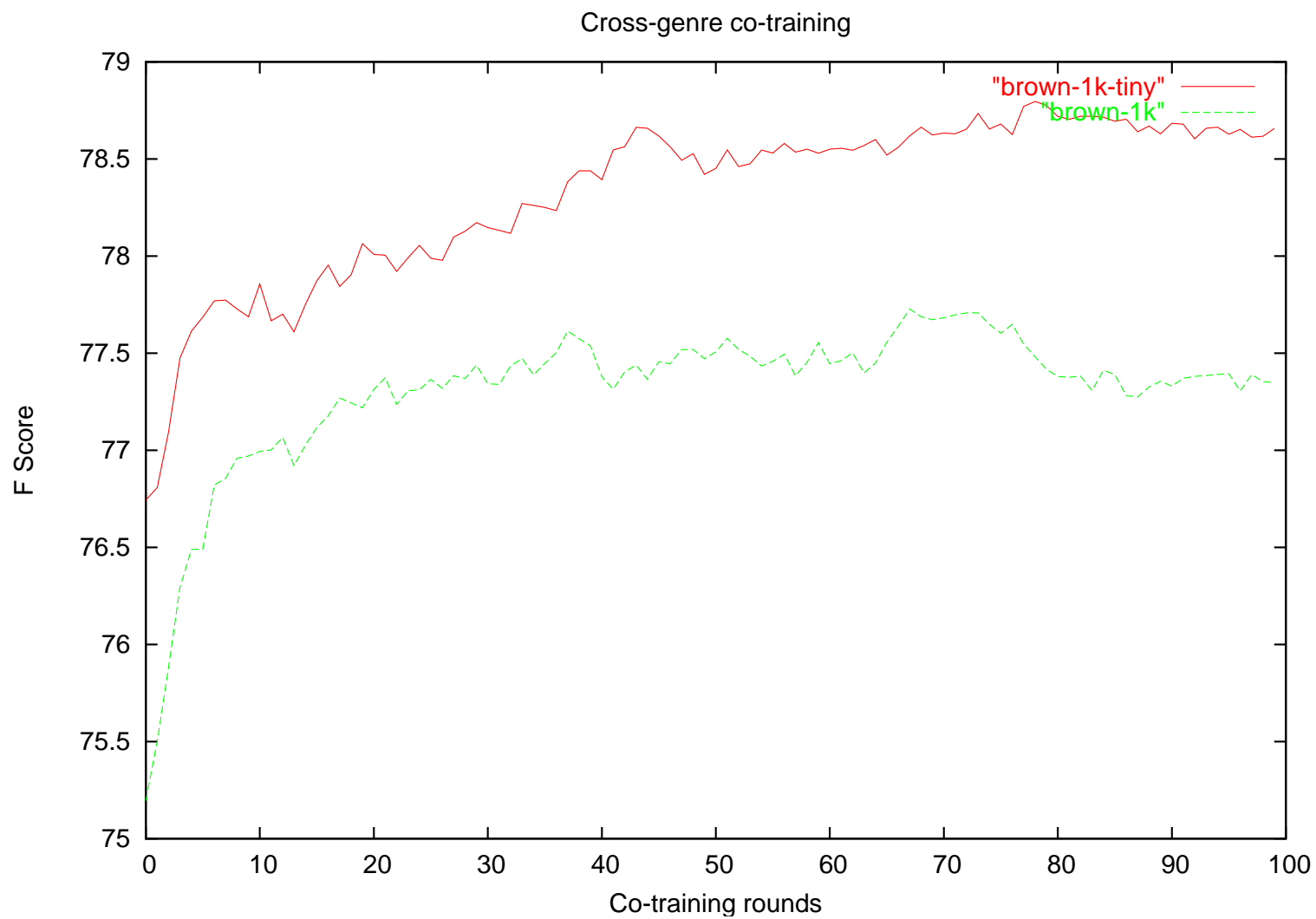
- Use co-training to boost performance, when faced with small seed data
 - Use 500 sentences of WSJ labelled data as seed data
 - Compare performance of co-training vs. self-training
- Use co-training to port parsers to new genres
 - Use Brown corpus as seed data, co-train and test on WSJ
- Use a large set of labelled data and use unlabelled data to improve parsing performance
 - Use Penn Treebank (40K sents) as seed data



- Use co-training to boost performance, when faced with small seed data
 - Co-training beats self-training with 500 sentence seed data
 - Compare performance when seed data is doubled to 1K sentences
- Use co-training to port parsers to new genres
 - Use Brown corpus as seed data, co-train and test on WSJ
- Use a large set of labeled data and use unlabeled data to improve parsing performance
 - Use Penn Treebank (40K sents) as seed data

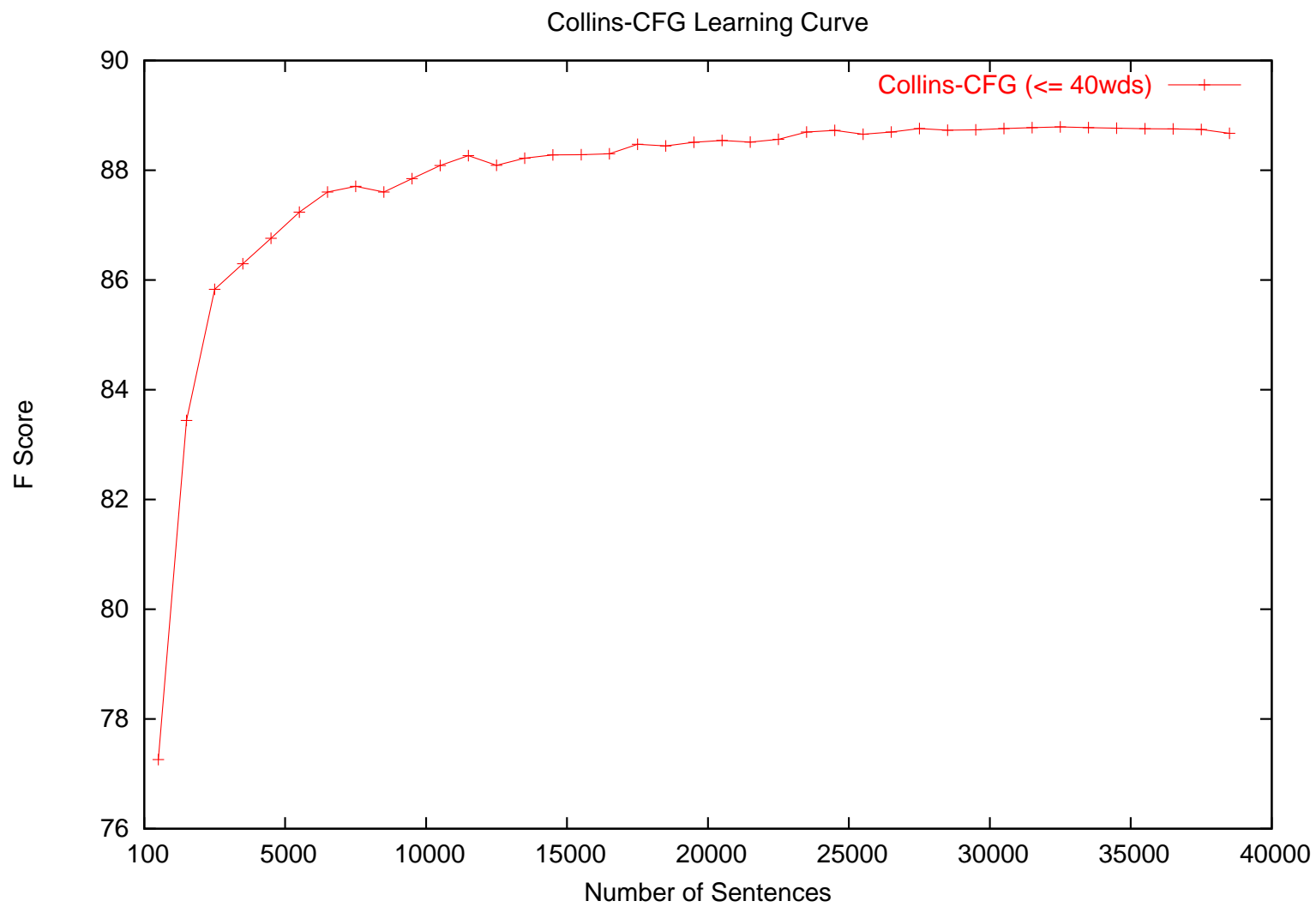


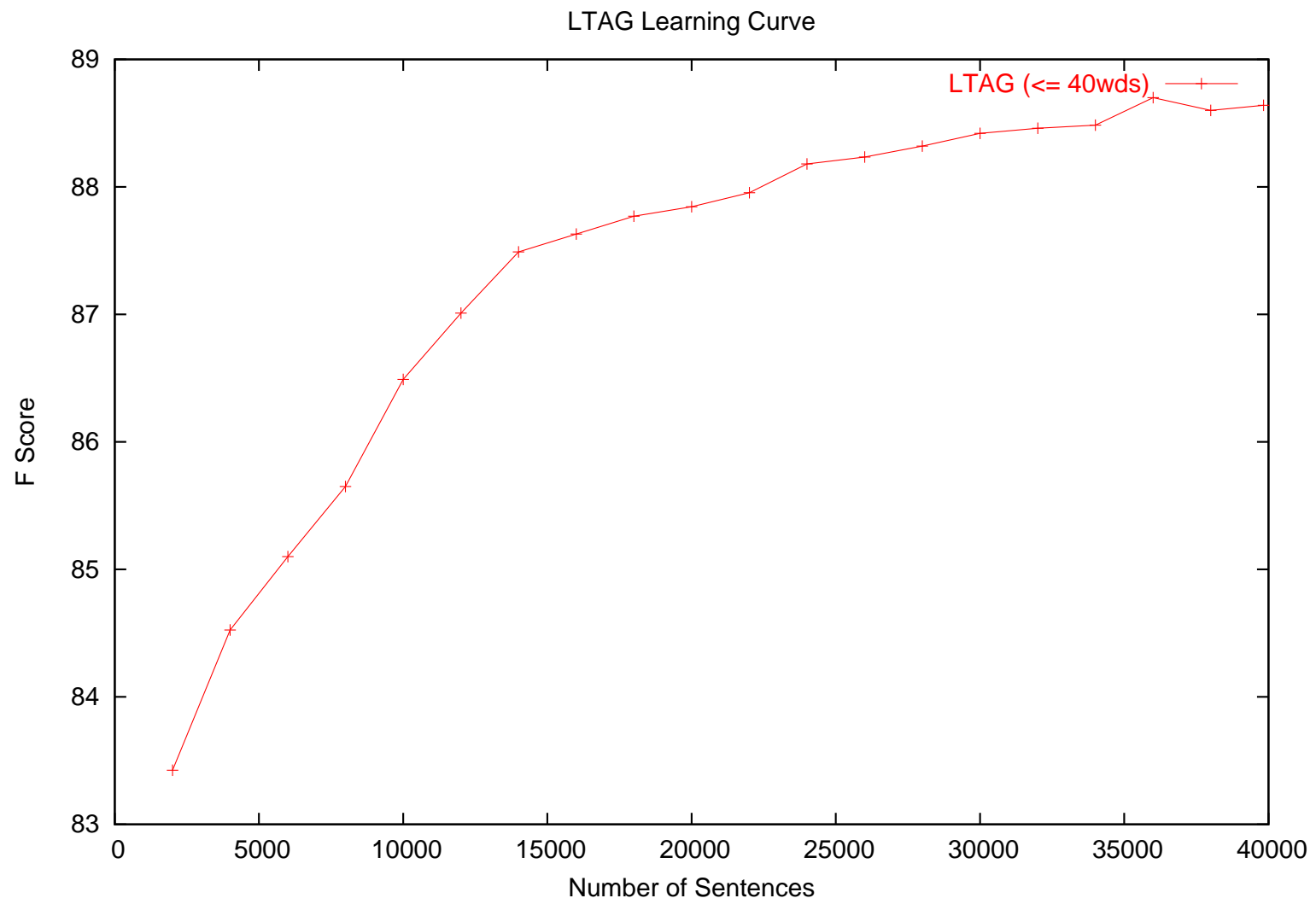
- Use co-training to boost performance, when faced with small seed data
 - Co-training beats self-training with 500 sentence seed data
 - Co-training still improves performance with 1K sentence seed data
- Use co-training to port parsers to new genres
 - Use Brown corpus as seed data, co-train and test on WSJ
- Use a large set of labeled data and use unlabeled data to improve parsing performance
 - Use Penn Treebank (40K sents) as seed data



- Use co-training to boost performance, when faced with small seed data
 - Co-training beats self-training with 500 sentence seed data
 - Different parse selection methods better for different parser views
 - Co-training still improves performance with 1K sentence seed data
- Use co-training to port parsers to new genres
 - Co-training improves performance significantly when porting from one genre (Brown) to another (WSJ)
- Use a large set of labeled data and use unlabeled data to improve parsing performance
 - Use Penn Treebank (40K sents) as seed data

- Experiments using 40K sentences Penn Treebank WSJ sentences as seed data for co-training did not produce a positive result
- Even after adding 260K sentences of unlabeled data using co-training did not significantly improve performance over the baseline
- However, we plan to do more experiments in the future which leverage more recent work on parse selection and the difference between the Collins-CFG and LTAG views





Summary

Experiment	Before(Sec 23)	After(Sec 23)
WSJ Self-training	74.4	74.3
WSJ (500) Co-training	74.4	76.9
WSJ (1k) Co-training	78.6	79.0
Brown co-training	73.6	76.8
Brown+ small WSJ co-training	75.4	78.2

- Use co-training to boost performance, when faced with small seed data
 - Co-training beats self-training with 500 sentence seed data
 - Co-training still improves performance with 1K sentence seed data
- Use co-training to port parsers to new genres
 - Co-training improves performance significantly when porting from one genre (Brown) to another (WSJ)
- Use a large set of labeled data and use unlabeled data to improve parsing performance
 - Using 40K sentences of Penn Treebank as seed data showed no improvement over the baseline. Future work: improving LTAG performance

- Acknowledgements

This work was done mostly during the NSF/DARPA JHU Language Engineering Summer Workshop 2002. This is joint work with R. Hwa, M. Osborne, M. Steedman, S. Clark, J. Hockenmaier, P. Ruhlen, S. Baker, J. Crim

- For more details about this work:

- Bootstrapping Statistical Parsers from Small Datasets: EACL 2003
- Example Selection for Bootstrapping Statistical Parsers: NAACL 2003