

# Co-training with Re-rankers

Jeremiah Crim\*  
University of Pennsylvania  
jcrim@seas.upenn.edu

November 25, 2003

## 1 Introduction

Researchers at the 2002 Center for Language and Speech Processing summer workshop explored the application of co-training to the task of statistical parsing. Under the co-training framework, the predictions of one parser on unlabelled text are used as new labelled examples to help re-train another parser. The parsers' roles are then reversed, and the process continues iteratively.

Initial small-scale experiments at the workshop showed that co-training can improve performance of parsers trained on small amounts (50-100 sentences) of labelled data. The workshop team was unable to report gains for parsers trained on the entire Wall Street Journal.

While the majority of the workshop was spent exploring the effects of co-training parsers, the team also began to evaluate the effect of co-training re-rankers. This work continues that project, examining the possible gain from co-training re-rankers.

---

\*supervised by Miles Osborne and Jason Baldridge at Univeristy of Edinburgh

## 2 Motivation

### 2.1 What is re-ranking?

A re-ranker reorders the output of a statistical parser. Given a set of parses for a sentence and a ranking of those parses, a re-ranker returns a new ranking for the parses.

### 2.2 Re-ranking vs. parsing

There are a number of reasons to consider co-training re-rankers instead of co-training parsers:

1. Performance

While most parsers use local features to help them distinguish between possible parses, rerankers can examine features that span the entire parse tree. This ability to consider extra features allows re-rankers to improve the performance of existing parsers. Collins (2000) cites a 13 percent reduction of error rate by re-ranking.

2. Speed

Experiments at the workshop suggest that a large amount of unlabelled data is necessary to get an improvement from co-training. But parsing is slow, and data must be parsed multiple times in order to

co-train parsers. If re-ranking, the unlabelled data only needs to be parsed once. It must be re-ranked several times, but this can be done very quickly.

### 3. Objective Function

Not all of the output from one statistical parser would be useful to help train another parser. The parser being used to label sentences may make mistakes, and adding too many incorrect parses to the training data of the other parser may hurt its accuracy rather than helping.

The co-training literature suggests that predictions of two classifiers will become increasingly correlated as their accuracy improves. Because of this, we would like to choose the unlabelled sentences that maximize agreement between the predictions of the parsers. This agreement metric requires retraining a parser multiple times on different subsets of the other parser's output in order to choose which newly labelled examples are useful. However, the long runtime of parsers forced the workshop team to come up with approximate measures for the agreement metric that did not require retraining a parser multiple times.

Because re-rankers can be trained so much more quickly than parsers, we can explicitly measure the objective function of agreement between re-rankers. No approximate measures will be necessary.

### 4. Task Closer to Classification

Co-training was originally introduced as a technique for improving binary classification. Re-rankers can be seen as binary classifiers: either a parse is the best one for a given sentence, or it isn't.

## 3 Data and Models

### 3.1 The re-rankers

The sentences were parsed using a Head-driven phrase structure grammar (HPSG) parser. A number of re-rankers were then trained. All re-rankers either used a log-linear model or a perceptron model to learn correlations between feature sets and preferred parses. Three feature sets were used:

1. "Ngram" features - variable-length ngrams over the flattened parse tree.
2. "Configurational" features - all possible subtrees of a parse tree.
3. "Semantic" features - a combination of:
  - a. nodes (subtrees of depth 1) from the phrase structure trees
  - b. parent/child relations on the semantic representation of the sentence

### 3.2 The data

The data consists of conversations about travel plans and business meetings.

## 4 Experiments

### 4.1 Baseline experiments

Initially, we re-ranked parses for a batch of 200 sentences with one re-ranker, and added the entire output to the training data for a second re-ranker. After retraining, the second re-ranker labelled for the first, and the process was iterated. Approximately 5000 sentences of data were added.

## 4.2 Approximate methods

In a second set of experiments, we added only a subset of the data labelled in each batch to the training data for the other re-ranker. The examples added were chosen based on the relative entropy of each sentence, based on the scores assigned to parses by the re-ranker that was currently labelling.

## 4.3 Maximizing agreement

As discussed earlier, we would like to choose which sentences to use as additional training data based not on measures such as entropy but on agreement between one re-ranker and the other retrained re-ranker. To do this, we:

1. Select  $n$  groups of sentences from the current batch.
2. Also select a set (A) of unlabelled data on which to measure agreement between the two re-rankers.
3. Label set A with both re-rankers - the “teacher” that will be labelling this round and the “learner” that we will retrain - and compute the degree to which the predictions of teacher and learner agree.
4. For each of the  $n$  groups of sentences:
  - a. Label the group with the teacher, and the temporarily retrain the learner based on these predictions.
  - b. Relabel set A with the retrained learner and compute the agreement between its predictions and the original predictions of the teacher.
5. From the  $n$  sets, choose the one that leads to the largest increase in agreement and permanently add this to the training data for the learner. If none of the sets lead to an improvement, don’t add anything this round.

6. If any training data was added, retrain the learner and switch roles.

Because a re-ranker can predict multiple best parses for a given sentence, computing agreement between the predictions of two re-rankers is not a straightforward task. If each re-ranker chooses only one best parse, the agreement for that sentence is one of the choice is the same for both re-rankers and 0 otherwise. When multiple parses are given the top rank by a re-ranker, there are many possible ways to assign an agreement score.

The simplest method would be to randomly choose from the best parses predicted by each re-ranker. If the random choice is the same, score 1 for agreement, and assign a score of 0 otherwise. But we prefer to use a metric that measures the overlap between the sets of best parses assigned by each re-ranker.

We use the following score for agreement:

$$\frac{(\text{no. of best parses agreed on})}{(\text{total no. of best parses proposed by either re-ranker})}$$

For example, if re-ranker 1 chose parses 1, 2 and 3 and re-ranker 2 picked parses 2, 3 and 4, then the agreement would be 2 out of 4 (.5) on the sentence.

## 4.4 Interpolation

We also used linear interpolation to combine the results of the original re-rankers with the results of the co-trained re-rankers at each iteration.

## 4.5 Using a held-out set

Initially, we used a held-out set to decide when to stop co-training.

We also considered using another held-out set to decide whether or not to add data to the learner at each round of co-training. Even if adding a set of data to a learner increased

agreement between re-rankers, we would only retrain on the set if it helped performance on a held-out set of data.

However, since holding out data decreases the amount of training material, which we already had a limited supply of, it seemed that simply training on this data to begin with was more useful than holding it out to decide which new data to retrain on.

## 4.6 Groups of learners

Until this point, all experiments carried out involved co-training two re-rankers, each on the output of the other. We also considered experiments in which the outputs of multiple (up to six) re-rankers were used to retrain each other.

We explored many choices for how to combine the outputs of multiple re-rankers:

1. Add only examples that all learners agree on to the pool of training data? (No - the re-rankers would only be training on a subset of their own output then.)
2. Add only examples where fewer than  $n$  learners disagree?
3. Vote on all unlabelled examples and add the combined predictions? This choice leads to more questions:
  - a. Vote on all ranks?
  - b. Vote on just the top rank?
  - c. Combine the raw scores assigned to each parse by the re-rankers (this makes sense if our re-rankers are log linear models, but not when we're using perceptrons)
4. Separate the re-rankers into different groups, and have each group label for the other groups?  
Intuitively, the re-rankers in each group

should be somehow different from those not in the group. This way, re-rankers in other groups will learn something they know from this group, but hopefully the fact that there are multiple re-rankers in this group making predictions together will cut down on the noise in their output.

Some groups we tried:

- a. Put all log linear re-rankers in one group and all perceptron models in another.
- b. Use three groups, each with both the log-linear and the perceptron model trained with a certain feature set.
- c. Assign each re-ranker to its own group.

## 4.7 Incremental learning

When considering which sentences to retrain the re-rankers on, we noted that our models were very good at labelling sentences with a small number of parses. The reason for this seems obvious: there is presumably less ambiguity in these sentences. However, unlike in some other corpora, sentences with only a few possible parses under the HPSG grammar do not necessarily lack information. Some quite long sentences will be assigned only a few parses - thus, these are easy to label, but contain many features that we can learn about from a single example.

To take advantage of this, we tried labelling only sentences with fewer than 3 parses and co-training with this output. Though this reduced the amount of unlabelled material significantly, we were able to show a slight increase with very little effort.

In future work, we hope to iterate this procedure, first learning from sentences with fewer than 3 parses, then interpolating these results with those obtained by labelling sen-

tences with fewer than 4 parses, and so on. We would probably want to lower the interpolation constant at each successive iteration, as our re-rankers tackle harder sentences.

## **5 Acknowledgements**

This material is based upon work supported by the National Science Foundation under Grant No. 0121285. The project was supervised by Jason Baldridge and Miles Osborne at the University of Edinburgh. Alex Lascarides also provided valuable advice about the semantics of HPSG parses. The work built on code written by Jason Baldridge for the ROSIE (Robust Semantic Interpretation) project at University of Edinburgh and Stanford University.