Example Selection for Bootstrapping Statistical Parsers

Mark Steedman[*], **Rebecca Hwa**[§], **Stephen Clark**[*], **Miles Osborne**[*], **Anoop Sarkar**[¶]
Julia Hockenmaier[*], **Paul Ruhlen**[†] **Steven Baker**[‡], **Jeremiah Crim**[†]

[*]School of Informatics, University of Edinburgh
{steedman,stephenc,julia,osborne}@cogsci.ed.ac.uk

[§]Institute for Advanced Computer Studies, University of Maryland
hwa@umiacs.umd.edu

[¶]School of Computing Science, Simon Fraser University
anoop@cs.sfu.ca

[†]Center for Language and Speech Processing, Johns Hopkins University
jcrim@jhu.edu,ruhlen@cs.jhu.edu

[‡]Department of Computer Science, Cornell University
sdb22@cornell.edu

Abstract

This paper investigates bootstrapping for statistical parsers to reduce their reliance on manually annotated training data. We consider both a mostly-unsupervised approach, *co-training*, in which two parsers are iteratively re-trained on each other's output; and a semi-supervised approach, *corrected co-training*, in which a human corrects each parser's output before adding it to the training data. The selection of labeled training examples is an integral part of both frameworks. We propose several selection methods based on the criteria of minimizing errors in the data and maximizing training utility. We show that incorporating the utility criterion into the selection method results in better parsers for both frameworks.

1 Introduction

Current state-of-the-art statistical parsers (Collins, 1999; Charniak, 2000) are trained on large annotated corpora such as the Penn Treebank (Marcus et al., 1993). However, the production of such corpora is expensive and labor-intensive. Given this bottleneck, there is considerable interest in (partially) automating the annotation process.

To overcome this bottleneck, two approaches from machine learning have been applied to training parsers. One is *sample selection* (Thompson et al., 1999; Hwa, 2000; Tang et al., 2002), a variant of active learning (Cohn et al., 1994), which tries to identify a small set of unlabeled sentences with high *training utility* for the human to label[1]. Sentences with high training utility are those most likely to improve the parser. The other approach, and the focus of this paper, is *co-training* (Sarkar, 2001), a mostly-unsupervised algorithm that replaces the human by having two (or more) parsers label training examples for each other. The goal is for both parsers to improve by bootstrapping off each other's strengths. Because the parsers may label examples incorrectly, only a subset of their output, chosen by some selection mechanism, is used in order to minimize errors. The choice of selection method significantly affects the quality of the resulting parsers.

We investigate a novel approach of selecting training examples for co-training parsers by incorporating the idea of maximizing training utility from sample selection. The selection mechanism is integral to both sample selection and co-training; however, because co-training and sample selection have different goals, their selection methods focus on different criteria: co-training typically favors selecting accurately labeled examples, while sample selection typically favors selecting examples with high training utility, which often are not sentences that the parsers already label accurately. In this work, we investigate selection methods for co-training that explore the trade-off between maximizing training utility and minimizing errors.

Empirical studies were conducted to compare selection methods under both co-training and a semi-supervised framework called *corrected co-training* (Pierce and Cardie, 2001), in which the selected examples are manually checked and corrected before being added to the

[1]In the context of training parsers, a labeled example is a sentence with its parse tree. Throughout this paper, we use the term "label" and "parse" interchangeably.

training data. For co-training, we show that the benefit of selecting examples with high training utility can offset the additional errors they contain. For corrected co-training, we show that selecting examples with high training utility reduces the number of sentences the human annotator has to check. For both frameworks, we show that selection methods that maximize training utility find labeled examples that result in better trained parsers than those that only minimize error.

2 Co-training

Blum and Mitchell (1998) introduced co-training to bootstrap two classifiers with different *views* of the data. The two classifiers are initially trained on a small amount of annotated seed data; then they label unannotated data for each other in an iterative training process. Blum and Mitchell prove that, when the two views are *conditionally independent* given the label, and each view is sufficient for learning the task, co-training can boost an initial weak learner using unlabeled data.

The theory underlying co-training has been extended by Dasgupta et al. (2002) to prove that, by maximizing their agreement over the unlabeled data, the two learners make few generalization errors (under the same independence assumption adopted by Blum and Mitchell). Abney (2002) argues that this assumption is extremely strong and typically violated in the data, and he proposes a weaker independence assumption.

Goldman and Zhou (2000) show that, through careful selection of newly labeled examples, co-training can work even when the classifiers' views do not satisfy the independence assumption. In this paper we investigate methods for selecting labeled examples produced by two statistical parsers. We do not explicitly maximize agreement (along the lines of Abney's algorithm (2002)) because it is too computationally intensive for training parsers.

The pseudocode for our co-training framework is given in Figure 1. It consists of two different parsers and a central control that interfaces between the two parsers and the data. At each co-training iteration, a small set of sentences is drawn from a large pool of unlabeled sentences and stored in a *cache*. Both parsers then attempt to label every sentence in the cache. Next, a subset of the newly labeled sentences is selected to be added to the training data. The examples added to the training set of one parser (referred to as the *student*) are only those produced by the other parser (referred to as the *teacher*), although the methods we use generalize to the case in which the parsers share a single training set. During selection, one parser first acts as the teacher and the other as the student, and then the roles are reversed.

A and B are two different parsers.
M_A^i and M_B^i are the models of A and B at step i.
U is a large pool of unlabeled sentences.
U^i is a small cache holding a subset of U at step i.
L is the manually labeled seed data.
L_A^i and L_B^i are the labeled training examples for A and B at step i.

Initialize:

$L_A^0 \leftarrow L_B^0 \leftarrow L$.
$M_A^0 \leftarrow Train(A, L_A^0)$
$M_B^0 \leftarrow Train(B, L_B^0)$

Loop:

$U^i \leftarrow$ Add unlabeled 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 $\{P_A\}$
$M_A^{i+1} \leftarrow Train(A, L_A^{i+1})$
$M_B^{i+1} \leftarrow Train(B, L_B^{i+1})$

Figure 1: The pseudo-code for the co-training algorithm

3 Selecting Training Examples

In each iteration, selection is performed in two steps. First, each parser uses some *scoring function*, f, to assess the parses it generated for the sentences in the cache.[2] Second, the central control uses some *selection method*, S, to choose a subset of these labeled sentences (based on the scores assigned by f) to add to the parsers' training data. The focus of this paper is on the selection phase, but to more fully investigate the effect of different selection methods we also consider two possible scoring functions.

3.1 Scoring functions

The scoring function attempts to quantify the correctness of the parses produced by each parser. An ideal scoring function would give the true accuracy rates (e.g., F-score, the combined labeled precision and recall rates). In practice, accuracy is approximated by some notion of confidence. For example, one easy-to-compute scoring function measures the conditional probability of the (most likely) parse. If a high probability is assigned, the parser is said to be confident in the label it produced.

In our experimental studies, we considered the selection methods' interaction with two scoring functions: an oracle scoring function $f_{\textit{F-score}}$ that returns the F-score of the parse as measured against a gold standard, and a

[2] In our experiments, both parsers use the same scoring function.

practical scoring function f_{prob} that returns the conditional probability of the parse.[3]

3.2 Selection methods

Based on the scores assigned by the scoring function, the selection method chooses a subset of the parser labeled sentences that best satisfy some selection criteria. One such criterion is the accuracy of the labeled examples, which may be estimated by the teacher parser's confidence in its labels. However, the examples that the teacher correctly labeled may not be those that the student needs. We hypothesize that the training utility of the examples for the student parser is another important criterion.

Training utility measures the improvement a parser would make if that sentence were correctly labeled and added to the training set. Like accuracy, the utility of an unlabeled sentence is difficult to quantify; therefore, we approximate it with values that can be computed from features of the sentence. For example, sentences containing many unknown words may have high training utility; so might sentences that a parser has trouble parsing. Under the co-training framework, we estimate the training utility of a sentence for the student by comparing the score the student assigned to its parse (according to its scoring function) against the score the teacher assigned to its own parse.

To investigate how the selection criteria of utility and accuracy affect the co-training process, we considered a number of selection methods that satisfy the requirements of accuracy and training utility to varying degrees. The different selection methods are shown below. For each method, a sentence (as labeled by the teacher parser) is selected if:

- ***above-n*** ($S_{above\text{-}n}$): the score of the teacher's parse (using its scoring function) $\geq n$.
- ***difference*** ($S_{diff\text{-}n}$): the score of the teacher's parse is greater than the score of the student's parse by some threshold n.
- ***intersection*** ($S_{int\text{-}n}$): the score of the teacher's parse is in the set of the teacher's n percent highest-scoring labeled sentences, and the score of the student's parse for the same sentence is in the set of the student's n percent lowest-scoring labeled sentences.

Each selection method has a *control parameter*, n, that determines the number of labeled sentences to add at each co-training iteration. It also serves as an indirect control of the number of errors added to the training set. For example, the $S_{above\text{-}n}$ method would allow more sentences to be selected if n was set to a low value (with respect to the scoring function); however, this is likely to reduce the accuracy rate of the training set.

The *above-n* method attempts to maximize the accuracy of the data (assuming that parses with higher scores are more accurate). The *difference* method attempts to maximize training utility: as long as the teacher's labeling is more accurate than that of the student, it is chosen, even if its absolute accuracy rate is low. The *intersection* method attempts to maximize both: the selected sentences are accurately labeled by the teacher *and* incorrectly labeled by the student.

4 Experiments

Experiments were performed to compare the effect of the selection methods on co-training and corrected co-training. We consider a selection method, S_1, superior to another, S_2, if, when a large unlabeled pool of sentences has been exhausted, the examples selected by S_1 (as labeled by the machine, and possibly corrected by the human) improve the parser more than those selected by S_2. All experiments shared the same general setup, as described below.

4.1 Experimental Setup

For two parsers to co-train, they should generate comparable output but use independent statistical models. In our experiments, we used a lexicalized context free grammar parser developed by Collins (1999), and a lexicalized Tree Adjoining Grammar parser developed by Sarkar (2002). Both parsers were initialized with some seed data. Since the goal is to minimize human annotated data, the size of the seed data should be small. In this paper we used a seed set size of $1,000$ sentences, taken from section 2 of the Wall Street Journal (WSJ) Penn Treebank. The total pool of unlabeled sentences was the remainder of sections 2-21 (stripped of their annotations), consisting of about 38,000 sentences. The cache size is set at 500 sentences. We have explored using different settings for the seed set size (Steedman et al., 2003).

The parsers were evaluated on unseen test sentences (section 23 of the WSJ corpus). Section 0 was used as a development set for determining parameters. The evaluation metric is the Parseval F-score over labeled constituents: $\textit{F-score} = \frac{2 \times LR \times LP}{LR+LP}$, where LP and LR are labeled precision and recall rate, respectively. Both parsers were evaluated, but for brevity, all results reported here are for the Collins parser, which received higher Parseval scores.

[3] A nice property of using conditional probability, $Pr(parse|sentence)$, as the scoring function is that it normalizes for sentence length.

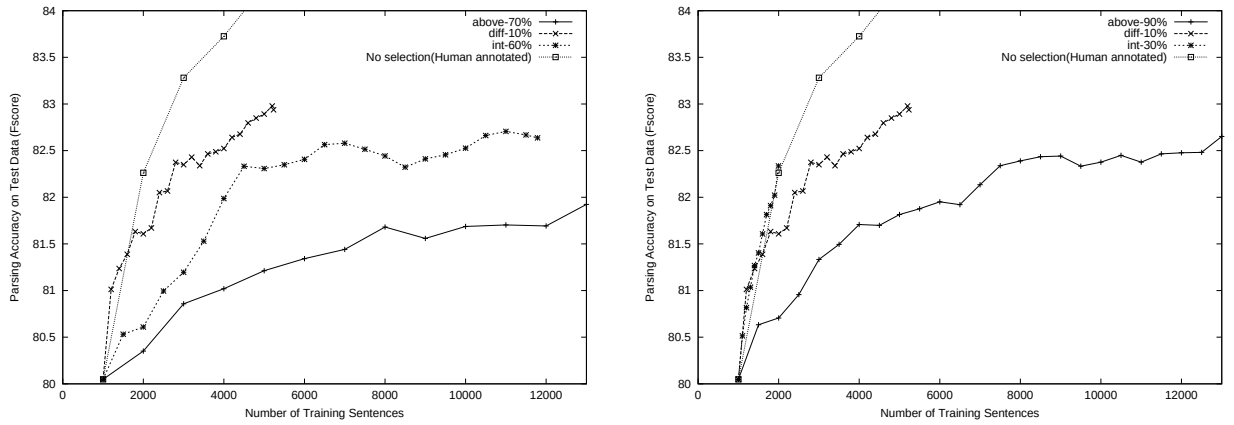


Figure 2: A comparison of selection methods using the oracle scoring function, $f_{F\text{-}score}$, controlling for the label quality of the training data. (a) The average accuracy rates are about 85%. (b) The average accuracy rates (except for those selected by $S_{diff\text{-}10\%}$) are about 95%.

4.2 Experiment 1: Selection Methods and Co-Training

We first examine the effect of the three selection methods on co-training without correction (i.e., the chosen machine-labeled training examples may contain errors). Because the selection decisions are based on the scores that the parsers assign to their outputs, the reliability of the scoring function has a significant impact on the performance of the selection methods. We evaluate the effectiveness of the selection methods using two scoring functions. In Section 4.2.1, each parser assesses its output with an *oracle* scoring function that returns the Parseval F-score of the output (as compared to the human annotated gold-standard). This is an idealized condition that gives us direct control over the error rate of the labeled training data. By keeping the error rates constant, our goal is to determine which selection method is more successful in finding sentences with high training utility.

In Section 4.2.2 we replace the oracle scoring function with f_{prob}, which returns the conditional probability of the best parse as the score. We compare how the selection methods' performances degrade under the realistic condition of basing selection decisions on unreliable parser output assessment scores.

4.2.1 Using the oracle scoring function, $f_{F\text{-}score}$

The goal of this experiment is to evaluate the selection methods using a reliable scoring function. We therefore use an oracle scoring function, $f_{F\text{-}score}$, which guarantees a perfect assessment of the parser's output. This, however, may be too powerful. In practice, we expect even a reliable scoring function to sometimes assign high scores to inaccurate parses. We account for this effect by adjusting the selection method's control parameter to affect two factors: the accuracy rate of the newly labeled training data, and the number of labeled sentences added at each training iteration. A relaxed parameter setting adds more parses to the training data, but also reduces the accuracy of the training data.

Figure 2 compares the effect of the three selection methods on co-training for the relaxed (left graph) and the strict (right graph) parameter settings. Each curve in the two graphs charts the improvement in the parser's accuracy in parsing the test sentences (y-axis) as it is trained on more data chosen by its selection method (x-axis). The curves have different endpoints because the selection methods chose a different number of sentences from the same 38K unlabeled pool. For reference, we also plotted the improvement of a fully-supervised parser (i.e., trained on human-annotated data, with no selection).

For the more relaxed setting, the parameters are chosen so that the newly labeled training data have an average accuracy rate of about 85%:

- $S_{above\text{-}70\%}$ requires the labels to have an F-score $\geq$ 70%. It adds about 330 labeled sentences (out of the 500 sentence cache) with an average accuracy rate of 85% to the training data per iteration.

- $S_{diff\text{-}10\%}$ requires the score difference between the teacher's labeling and the student's labeling to be at least 10%. It adds about 50 labeled sentences with an average accuracy rate of 80%.

- $S_{int\text{-}60\%}$ requires the teacher's parse to be in the top 60% of its output and the student's parse for the same sentence to be in its bottom 60%. It adds about

150 labeled sentences with an average accuracy rate of 85%.

Although none rivals the parser trained on human annotated data, the selection method that improves the parser the most is $S_{diff\text{-}10\%}$. One interpretation is that the training utility of the examples chosen by $S_{diff\text{-}10\%}$ outweighs the cost of errors introduced into the training data. Another interpretation is that the other two selection methods let in too many sentences containing errors. In the right graph, we compare the same $S_{diff\text{-}10\%}$ with the other two selection methods using stricter control, such that the average accuracy rate for these methods is now about 95%:

- $S_{above\text{-}90\%}$ now requires the parses to be at least 90% correct. It adds about 150 labeled sentences per iteration.
- $S_{int\text{-}30\%}$ now requires the teacher's parse to be in the top 30% of its output and the student's parse for the same sentence in its bottom 30%. It adds about 15 labeled sentences.

The stricter control on $S_{above\text{-}90\%}$ improved the parser's performance, but not enough to overtake $S_{diff\text{-}10\%}$ after all the sentences in the unlabeled pool had been considered, even though the training data of $S_{diff\text{-}10\%}$ contained many more errors. $S_{int\text{-}30\%}$ has a faster initial improvement[4], closely tracking the progress of the fully-supervised parser. However, the stringent requirement exhausted the unlabeled data pool before training the parser to convergence. $S_{int\text{-}30\%}$ might continue to help the parser to improve if it had access to more unlabeled data, which is easier to acquire than annotated data[5].

Comparing the three selection methods under both strict and relaxed control settings, the results suggest that training utility is an important criterion in selecting training examples, even at the cost of reduced accuracy.

4.2.2 Using the f_{prob} scoring function

To determine the effect of unreliable scores on the selection methods, we replace the oracle scoring function, $f_{F\text{-}score}$, with f_{prob}, which approximates the accuracy of a parse with its conditional probability. Although this is a poor estimate of accuracy (especially when computed from a partially trained parser), it is very easy to compute. The unreliable scores also reduce the correlation between the selection control parameters and the level of errors in the training data. In this experiment, we set the parameters for all three selection methods so that approximately 30-50 sentences were added to the training data per iteration. The average accuracy rate of the training data for $S_{above\text{-}70\%}$ was about 85%, and the rate for $S_{diff\text{-}30\%}$ and $S_{int\text{-}30\%}$ was about 75%.

[4]A fast improvement rate is not a central concern here, but it will be more relevant for corrected co-training.

[5]This oracle experiment is bounded by the size of the annotated portion of the WSJ corpus.

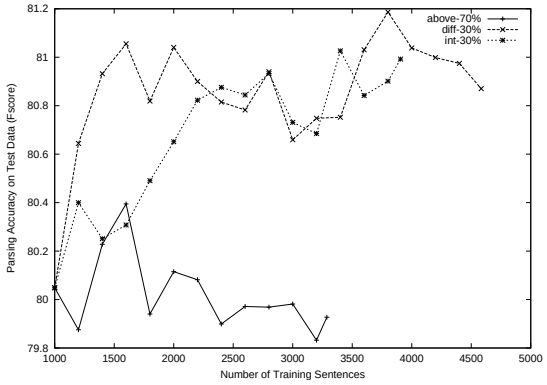


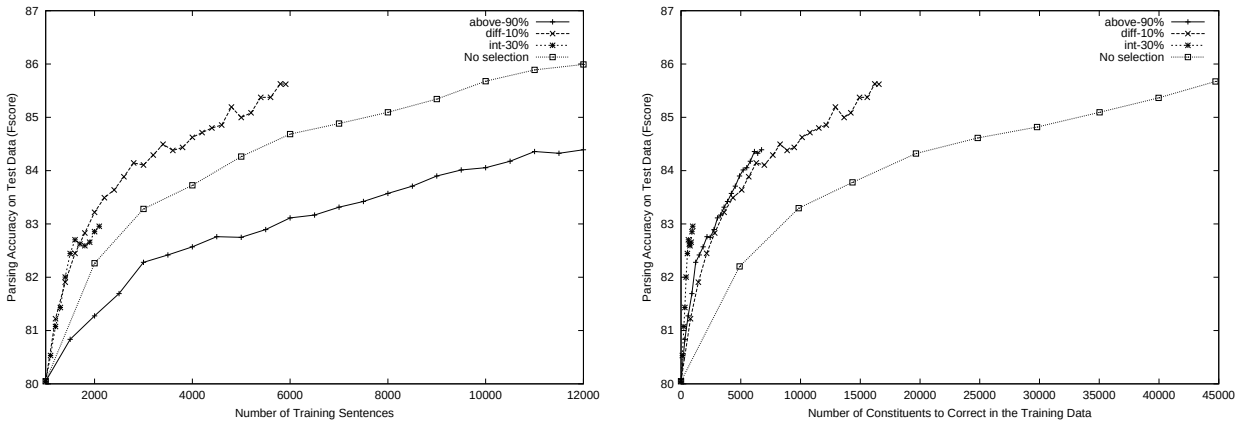
Figure 3: A comparison of selection methods using the conditional probability scoring function, f_{prob}.

As expected, the parser performances of all three selection methods using f_{prob} (shown in Figure 3) are lower than using $f_{F\text{-}score}$ (see Figure 2). However, $S_{diff\text{-}30\%}$ and $S_{int\text{-}30\%}$ helped the co-training parsers to improve with a 5% error reduction (1% absolute difference) over the parser trained only on the initial seed data. In contrast, despite an initial improvement, using $S_{above\text{-}70\%}$ did not help to improve the parser. In their experiments on NP identifiers, Pierce and Cardie (2001) observed a similar effect. They hypothesize that co-training does not scale well for natural language learning tasks that require a huge amount of training data because too many errors are accrued over time. Our experimental results suggest that the use of training utility in the selection process can make co-training parsers more tolerant to these accumulated errors.

4.3 Experiment 2: Selection Methods and Corrected Co-training

To address the problem of the training data accumulating too many errors over time, Pierce and Cardie proposed a semi-supervised variant of co-training called corrected co-training, which allows a human annotator to review and correct the output of the parsers before adding it to the training data. The main selection criterion in their co-training system is accuracy (approximated by confidence). They argue that selecting examples with nearly correct labels would require few manual interventions from the annotator.

We hypothesize that it may be beneficial to consider the training utility criterion in this framework as well. We perform experiments to determine whether selecting fewer (and possibly less accurately labeled) exam-



(a) (b)

Figure 4: A comparison of selection methods for corrected co-training using $f_{F\text{-}score}$ (a) in terms of the number of sentences added to the training data; (b) in terms of the number of manually corrected constituents.

ples with higher training utility would require less effort from the annotator. In our experiments, we simulated the interactive sample selection process by revealing the gold standard. As before, we compare the three selection methods using both $f_{F\text{-}score}$ and f_{prob} as scoring functions.[6]

4.3.1 Using the oracle scoring function, $f_{F\text{-}score}$

Figure 4 shows the effect of the three selection methods (using the strict parameter setting) on corrected co-training. As a point of reference, we plot the improvement rate for a fully supervised parser (same as the one in Figure 2). In addition to charting the parser's performance in terms of the number of labeled training sentences (left graph), we also chart the parser's performance in terms of the the number of constituents the machine mislabeled (right graph). The pair of graphs indicates the amount of human effort required: the left graph shows the number of sentences the human has to check, and the right graph shows the number of constituents the human has to correct.

Comparing $S_{above\text{-}90\%}$ and $S_{diff\text{-}10\%}$, we see that $S_{diff\text{-}10\%}$ trains a better parser than $S_{above\text{-}90\%}$ when all the unlabeled sentences have been considered. It also improves the parser using a smaller set of training examples. Thus, for the same parsing performance, it requires the human to check fewer sentences than $S_{above\text{-}90\%}$ and the reference case of no selection (Figure 4(a)). On the other hand, because the labeled sentences selected by $S_{diff\text{-}10\%}$ contain more mistakes than those selected by $S_{above\text{-}90\%}$, $S_{diff\text{-}10\%}$ requires slightly more corrections than $S_{above\text{-}90\%}$ for the same level of parsing performance; though both require fewer corrections than the reference case of no selection (Figure 4(b)). Because the amount of effort spent by the annotator depends on the number of sentences checked as well as the amount of corrections made, whether $S_{diff\text{-}10\%}$ or $S_{above\text{-}90\%}$ is more effort reducing may be a matter of the annotator's preference.

The selection method that improves the parser at the fastest rate is $S_{int\text{-}30\%}$. For the same parser performance level, it selects the fewest number of sentences for a human to check and requires the human to make the least number of corrections. However, as we have seen in the earlier experiment, very few sentences in the unlabeled pool satisfy its stringent criteria, so it ran out of data before the parser was trained to convergence. At this point we cannot determine whether $S_{int\text{-}30\%}$ might continue to improve the parser if we used a larger set of unlabeled data.

4.3.2 Using the f_{prob} scoring function

We also consider the effect of unreliable scores in the corrected co-training framework. A comparison between the selection methods using f_{prob} is reported in Figure 5. The left graph charts parser performance in terms of the number of sentences the human must check; the right charts parser performance in terms of the number of constituents the human must correct. As expected, the unreliable scoring function degrades the effectiveness of the selection methods; however, compared to its unsupervised counterpart (Figure 3), the degradation is not as severe. In fact, $S_{diff\text{-}30\%}$ and $S_{int\text{-}30\%}$ still require fewer training data than the reference parser. Moreover, consistent with the other experiments, the selection methods that attempt to maximize training utility achieve better parsing

[6] The selection control parameters are the same as the previous set of experiments, using the strict setting (i.e., Figure 2(b)) for $f_{F\text{-}score}$.

performance than $S_{above\text{-}70\%}$. Finally, in terms of reducing human effort, the three selection methods require the human to correct comparable amount of parser errors for the same level of parsing performance, but for $S_{diff\text{-}30\%}$ and $S_{int\text{-}30\%}$, fewer sentences need to be checked.

4.3.3 Discussion

Corrected co-training can be seen as a form of active learning, whose goal is to identify the smallest set of unlabeled data with high training utility for the human to label. Active learning can be applied to a single learner (Lewis and Catlett, 1994) and to multiple learners (Freund et al., 1997; Engelson and Dagan, 1996; Ngai and Yarowsky, 2000). In the context of parsing, all previous work (Thompson et al., 1999; Hwa, 2000; Tang et al., 2002) has focussed on single learners. Corrected co-training is the first application of active learning for multiple parsers. We are currently investigating comparisons to the single learner approaches.

Our approach is similar to *co-testing* (Muslea et al., 2002), an active learning technique that uses two classifiers to find contentious examples (i.e., data for which the classifiers' labels disagree) for a human to label. There is a subtle but significant difference, however, in that their goal is to reduce the total number of labeled training examples whereas we also wish to reduce the number of *corrections* made by the human. Therefore, our selection methods must take into account the quality of the parse produced by the teacher in addition to how different its parse is from the one produced by the student. The *intersection* method precisely aims at selecting sentences that satisfy both requirements. Exploring different selection methods is part of our on-going research effort.

5 Conclusion

We have considered three selection methods that have different priorities in balancing the two (often competing) criteria of accuracy and training utility. We have empirically compared their effect on co-training, in which two parsers label data for each other, as well as corrected co-training, in which a human corrects the parser labeled data before adding it to the training set. Our results suggest that training utility is an important selection criterion to consider, even at the cost of potentially reducing the accuracy of the training data. In our empirical studies, the selection method that aims to maximize training utility, $S_{diff\text{-}n}$, consistently finds better examples than the one that aims to maximize accuracy, $S_{above\text{-}n}$. Our results also suggest that the selection method that aims to maximize both accuracy and utility, $S_{int\text{-}n}$, shows promise in improving co-training parsers and in reducing human effort for corrected co-training; however, a much larger unlabeled data set is needed to verify the benefit of $S_{int\text{-}n}$.

The results of this study indicate the need for scoring functions that are better estimates of the accuracy of the parser's output than conditional probabilities. Our oracle experiments show that, by using effective selection methods, the co-training process can improve parser peformance even when the newly labeled parses are not completely accurate. This suggests that co-training may still be beneficial when using a practical scoring function that might only coarsely distinguish accurate parses from inaccurate parses. Further avenues to explore include the development of selection methods to efficiently approximate maximizing the objective function of parser agreement on unlabeled data, following the work of Dasgupta et al. (2002) and Abney (2002). Also, co-training might be made more effective if partial parses were used as training data. Finally, we are conducting experiments to compare corrected co-training with other active learning methods. We hope these studies will reveal ways to combine the strengths of co-training and active learning to make better use of unlabeled data.

Acknowledgments

This work has been supported, in part, by NSF/DARPA funded 2002 Human Language Engineering Workshop at JHU, EPSRC grant GR/M96889, the Department of Defense contract RD-02-5700, and ONR MURI Contract FCPO.810548265. We would like to thank Chris Callison-Burch, Michael Collins, John Henderson, Lillian Lee, Andrew McCallum, and Fernando Pereira for helpful discussions; to Ric Crabbe, Adam Lopez, the participants of CS775 at Cornell University, and the reviewers for their comments on this paper.

References

Steven Abney. 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367, Philadelphia, PA.

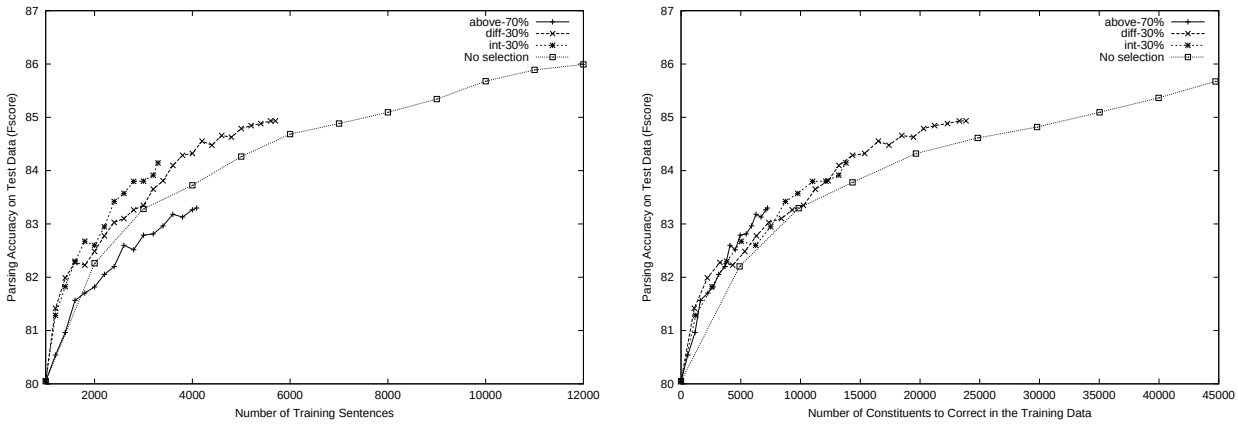
Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Annual Meeting of the NAACL*.

David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Sanjoy Dasgupta, Michael Littman, and David McAllester. 2002. PAC generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances*



(a) (b)

Figure 5: A comparison of selection methods for corrected co-training using f_{prob} (a) in terms of the number of sentences added to the training data; (b) in terms of the number of manually corrected constituents.

in Neural Information Processing Systems 14, Cambridge, MA. MIT Press.

Sean P. Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from copora. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 319–326.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168.

Sally Goldman and Yan Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA.

Rebecca Hwa. 2000. Sample selection for statistical grammar induction. In *Proceedings of the 2000 Joint SIGDAT Conference on EMNLP and VLC*, pages 45–52, Hong Kong, China, October.

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ion Muslea, Steve Minton, and Craig Knoblock. 2002. Selective sampling with redundant views. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 621–626.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 117–125, Hong Kong, China, October.

David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proceedings of the Empirical Methods in NLP Conference*, Pittsburgh, PA.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of the 2nd Annual Meeting of the NAACL*, pages 95–102, Pittsburgh, PA.

Anoop Sarkar. 2002. *Statistical Parsing Algorithms for Lexicalized Tree Adjoining Grammars*. Ph.D. thesis, University of Pennsylvania.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *The Proceedings of the Annual Meeting of the European Chapter of the ACL*. To appear.

Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 120–127, July.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of ICML-99*, pages 406–414, Bled, Slovenia.