Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets

Roi Reichart
ICNC
Hebrew University of Jerusalem
roiri@cs.huji.ac.il

Ari Rappoport
Institute of Computer Science
Hebrew University of Jerusalem
arir@cs.huji.ac.il

Abstract

Creating large amounts of annotated data to train statistical PCFG parsers is expensive, and the performance of such parsers declines when training and test data are taken from different domains. In this paper we use self-training in order to improve the quality of a parser and to adapt it to a different domain, using only small amounts of manually annotated seed data. We report significant improvement both when the seed and test data are in the same domain and in the out-of-domain adaptation scenario. In particular, we achieve 50% reduction in annotation cost for the in-domain case, yielding an improvement of 66% over previous work, and a 20-33% reduction for the domain adaptation case. This is the first time that self-training with small labeled datasets is applied successfully to these tasks. We were also able to formulate a characterization of when self-training is valuable.

1 Introduction

State of the art statistical parsers (Collins, 1999; Charniak, 2000; Koo and Collins, 2005; Charniak and Johnson, 2005) are trained on manually annotated treebanks that are highly expensive to create. Furthermore, the performance of these parsers decreases as the distance between the genres of their training and test data increases. Therefore, enhancing the performance of parsers when trained on *small* manually annotated datasets is of great importance, both when the seed and test data are taken from the same domain (the *in-domain* scenario) and when they are taken from different domains (the *out-of-domain* or *parser adaptation* scenario). Since the problem is the expense in manual annotation, we define 'small' to be 100-2,000 sentences, which are the sizes of sentence sets that can be manually annotated by constituent structure in a few hours[1].

Self-training is a method for using unannotated data when training supervised models. The model is first trained using manually annotated ('seed') data, then the model is used to automatically annotate a pool of unannotated ('self-training') data, and then the manually and automatically annotated datasets are combined to create the training data for the final model. Self-training of parsers trained on small datasets is of enormous potential practical importance, due to the huge amounts of unannotated data that are becoming available today and to the high cost of manual annotation.

In this paper we use self-training to enhance the performance of a generative statistical PCFG parser (Collins, 1999) for both the in-domain and the parser adaptation scenarios, using only small amounts of manually annotated data. We perform four experiments, examining all combinations of in-domain and out-of-domain seed and self-training data.

Our results show that self-training is of substantial benefit for the problem. In particular, we present:

- 50% reduction in annotation cost when the seed and test data are taken from the same domain, which is 66% higher than any previous result with small manually annotated datasets.

[1] We note in passing that quantitative research on the cost of annotation using various annotation schemes is clearly lacking.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 616–623,
Prague, Czech Republic, June 2007.

- The first time that self-training improves a generative parser when the seed and test data are from the same domain.

- 20-33% reduction in annotation cost when the seed and test data are from different domains.

- The first time that self-training succeeds in adapting a generative parser between domains using a small manually annotated dataset.

- The first formulation (related to the number of unknown words in a sentence) of when self-training is valuable.

Section 2 discusses previous work, and Section 3 compares in-depth our protocol to a previous one. Sections 4 and 5 present the experimental setup and our results, and Section 6 analyzes the results in an attempt to shed light on the phenomenon of self-training.

2 Related Work

Self-training might seem a strange idea: why should a parser trained on its own output learn anything new? Indeed, (Clark et al., 2003) applied self-training to POS-tagging with poor results, and (Charniak, 1997) applied it to a generative statistical PCFG parser trained on a large seed set (40K sentences), without any gain in performance.

Recently, (McClosky et al., 2006a; McClosky et al., 2006b) have successfully applied self-training to various parser adaptation scenarios using the reranking parser of (Charniak and Johnson, 2005). A reranking parser (see also (Koo and Collins, 2005)) is a layered model: the base layer is a generative statistical PCFG parser that creates a ranked list of k parses (say, 50), and the second layer is a reranker that reorders these parses using more detailed features. McClosky et al (2006a) use sections 2-21 of the WSJ PennTreebank as seed data and between 50K to 2,500K unlabeled NANC corpus sentences as self-training data. They train the PCFG parser and the reranker with the manually annotated WSJ data, and parse the NANC data with the 50-best PCFG parser. Then they proceed in two directions. In the first, they reorder the 50-best parse list with the reranker to create a new 1-best list. In the second, they leave the 1-best list produced by the generative PCFG parser untouched. Then they combine the 1-best list (each direction has its own list) with the WSJ training set, to retrain the PCFG parser. The final PCFG model and the reranker (trained only on annotated WSJ material) are then used to parse the test section (23) of WSJ.

There are two major differences between these papers and the current one, stemming from their usage of a reranker and of large seed data. First, when their 1-best list of the base PCFG parser was used as self training data for the PCFG parser (the second direction), the performance of the base parser did not improve. It had improved only when the 1-best list of the *reranker* was used. In this paper we show how the 1-best list of a base (generative) PCFG parser can be used as a self-training material for the base parser itself and enhance its performance, without using any reranker. This reveals a noteworthy characteristic of generative PCFG models and offers a potential direction for parser improvement, since the quality of a parser-reranker combination critically depends on that of the base parser.

Second, these papers did not explore self-training when the seed is small, a scenario whose importance has been discussed above. In general, PCFG models trained on small datasets are less likely to parse the self-training data correctly. For example, the f-score of WSJ data parsed by the base PCFG parser of (Charniak and Johnson, 2005) when trained on the training sections of WSJ is between 89% to 90%, while the f-score of WSJ data parsed with the Collins' model that we use, and a small seed, is between 40% and 80%. As a result, the good results of (McClosky et al, 2006a; 2006b) with large seed sets do not immediately imply success with small seed sets. Demonstration of such success is a contribution of the present paper.

Bacchiani et al (2006) explored the scenario of out-of-domain seed data (the Brown training set containing about 20K sentences) and in-domain self-training data (between 4K to 200K sentences from the WSJ) and showed an improvement over the baseline of training the parser with the seed data only. However, they did not explore the case of small seed datasets (the effort in manually annotating 20K is substantial) and their work addresses only one of our scenarios (OI, see below).

A work closely related to ours is (Steedman et al., 2003a), which applied co-training (Blum and Mitchell, 1998) and self-training to Collins' parsing model using a small seed dataset (500 sentences for both methods and 1,000 sentences for co-training only). The seed, self-training and test datasets they used are similar to those we use in our II experiment (see below), but the self-training protocols are different. They first train the parser with the seed sentences sampled from WSJ sections 2-21. Then, iteratively, 30 sentences are sampled from these sections, parsed by the parser, and the 20 best sentences (in terms of parser confidence defined as probability of top parse) are selected and combined with the previously annotated data to retrain the parser. The co-training protocol is similar except that each parser is trained with the 20 best sentences of the other parser. Self-training did not improve parser performance on the WSJ test section (23). Steedman et al (2003b) followed a similar co-training protocol except that the selection function (three functions were explored) considered the differences between the confidence scores of the two parsers. In this paper we show a self-training protocol that achieves better results than all of these methods (Table 2). The next section discusses possible explanations for the difference in results. Steedman et al (2003b) and Hwa et al, (2003) also used several versions of corrected co-training which are not comparable to ours and other suggested methods because their evaluation requires different measures (e.g. reviewed and corrected constituents are separately counted).

As far as we know, (Becker and Osborne, 2005) is the only additional work that tries to improve a generative PCFG parsers using small seed data. The techniques used are based on active learning (Cohn et al., 1994). The authors test two novel methods, along with the tree entropy (TE) method of (Hwa, 2004). The seed, the unannotated and the test sets, as well as the parser used in that work, are similar to those we use in our II experiment. Our results are superior, as shown in Table 3.

3 Self-Training Protocols

There are many possible ways to do self-training. A main goal of this paper is to identify a self-training protocol most suitable for enhancement and domain adaptation of statistical parsers trained on small datasets. No previous work has succeeded in identifying such a protocol for this task. In this section we try to understand why.

In the protocol we apply, the self-training set contains several thousand sentences A parser trained with a small seed set parses the self-training set, and then the *whole* automatically annotated self-training set is combined with the manually annotated seed set to retrain the parser. This protocol and that of Steedman et al (2003a) were applied to the problem, with the same seed, self-training and test sets. As we show below (see Section 4 and Section 5), while Steedman's protocol does not improve over the baseline of using only the seed data, our protocol does.

There are four differences between the protocols. First, Steedman et al's seed set consists of *consecutive* WSJ sentences, while we select them randomly. In the next section we show that this difference is immaterial. Second, Steedman et al's protocol looks for sentences of high quality parse, while our protocol prefers to use many sentences without checking their parse quality. Third, their protocol is iterative while ours uses a single step. Fourth, our self-training set is orders of magnitude larger than theirs. To examine the parse quality issue, we performed their experiment using their setting but selecting the high quality parse sentences using their f-score relative to the gold standard annotation from secs 2-21 rather than a quality estimate. No improvement over the baseline was achieved even with this oracle. Thus the problem with their protocol does not lie with the parse quality assessment function; no other function would produce results better than the oracle. To examine the iteration issue, we performed their experiment in a single step, selecting at once the oracle-best 2,000 among 3,000 sentences[2], which produced only a mediocre improvement. We thus conclude that the size of the self-training set is a major factor responsible for the difference between the protocols.

4 Experimental Setup

We used a reimplementation of Collins' parsing model 2 (Bikel, 2004). We performed four experiments, II, IO, OI, and OO, two with in-domain seed

[2] Corresponding to a 100 iterations of 30 sentences each.

(II, IO) and two with out-of-domain seed (OI, OO), examining in-domain self-training (II, OI) and out-of-domain self-training (IO, OO). Note that being ‘in’ or ‘out’ of domain is determined by the *test* data. Each experiment contained 19 runs. In each run a different seed size was used, from 100 sentences onwards, in steps of 100. For statistical significance, we repeated each experiment five times, in each repetition randomly sampling different manually annotated sentences to form the seed dataset[3].

The seed data were taken from WSJ sections 2-21. For II and IO, the test data is WSJ section 23 (2416 sentences) and the self-training data are either WSJ sections 2-21 (in II, excluding the seed sentences) or the Brown training section (in IO). For OI and OO, the test data is the Brown test section (2424 sentences), and the self-training data is either the Brown training section (in OI) or WSJ sections 2-21 (in OO). We removed the manual annotations from the self-training sections before using them.

For the Brown corpus, we based our division on (Bacchiani et al., 2006; McClosky et al., 2006b). The test and training sections consist of sentences from all of the genres that form the corpus. The training division consists of 90% (9 of each 10 consecutive sentences) of the data, and the test section are the remaining 10% (We did not use any held out data). Parsing performance is measured by f-score, $f = \frac{2 \times P \times R}{P+R}$, where P, R are labeled precision and recall.

To further demonstrate our results for parser adaptation, we also performed the OI experiment where seed data is taken from WSJ sections 2-21 and both self-training and test data are taken from the Switchboard corpus. The distance between the domains of these corpora is much greater than the distance between the domains of WSJ and Brown. The Brown and Switchboard corpora were divided to sections in the same way.

We have also performed all four experiments with the seed data taken from the Brown training section. The results were very similar and will not be detailed here due to space constraints.

5 Results

5.1 In-domain seed data

In these two experiments we show that when the seed and test data are taken from the same domain, a very significant enhancement of parser performance can be achieved, whether the self-training material is in-domain (II) or out-of-domain (IO). Figure 1 shows the improvement in parser f-score when self-training data is used, compared to when it is not used. Table 1 shows the reduction in manually annotated seed data needed to achieve certain f-score levels. The enhancement in performance is very impressive in the in-domain self-training data scenario – a reduction of 50% in the number of manually annotated sentences needed for achieving 75 and 80 f-score values. A significant improvement is achieved in the out-of-domain self-training scenario as well.

Table 2 compares our results with self-training and co-training results reported by (Steedman et al, 20003a; 2003b). As stated earlier, the experimental setup of these works is similar to ours, but the self-training protocols are different. For self-training, our II improves an absolute 3.74% over their 74.3% result, which constitutes a 14.5% reduction in error (from 25.7%).

The table shows that for both seed sizes our self training protocol outperforms both the self-training and co-training protocols of (Steedman et al, 20003a; 2003b). Results are not included in the table only if they are not reported in the relevant paper. The self-training protocol of (Steedman et al., 2003a) does not actually improve over the baseline of using only the seed data. Section 3 discussed a possible explanation to the difference in results.

In Table 3 we compare our results to the results of the methods tested in (Becker and Osborne, 2005) (including TE)[4]. To do that, we compare the reduction in manually annotated data needed to achieve an f-score value of 80 on WSJ section 23 achieved by each method. We chose this measure since it is

[3] (Steedman et al., 2003a) used the *first* 500 sentences of WSJ training section as seed data. For direct comparison, we performed our protocol in the II scenario using the first 500 or 1000 sentences of WSJ training section as seed data and got similar results to those reported below for our protocol with *random* selection. We also applied the protocol of Steedman et al to scenario II with 500 randomly selected sentences, getting no improvement over the random baseline.

[4] The measure is constituents and not sentences because this is how results are reported in (Becker and Osborne, 2005). However, the same reduction is obtained when sentences are counted, because the number of constituents is averaged when taking many sentences.

f-score	75	80
Seed data only	600(0%)	1400(0%)
II	**300**(**50**%)	**700**(**50**%)
IO	500(17%)	1200(14.5%)

Table 1: Number of in-domain seed sentences needed for achieving certain f-scores. Reductions compared to no self-training (line 1) are given in parentheses.

Seed size	our II	our IO	Steedman ST	Steedman CT 2003a	Steedman CT 2003b
500 sent.	**78.04**	75.81	74.3	76.9	—-
1,000 sent.	**81.43**	79.49	—-	79	81.2

Table 2: F-scores of our in-domain-seed self-training vs. self-training (ST) and co-training (CT) of (Steedman et al, 20003a; 2003b).

the only explicitly reported number in that work. As the table shows, our method is superior: our reduction of 50% constitutes an improvement of 66% over their best reduction of 30.6%.

When applying self-training to a parser trained with a small dataset we expect the coverage of the parser to increase, since the combined training set should contain items that the seed dataset does not. On the other hand, since the accuracy of annotation of such a parser is poor (see the no self-training curve in Figure 1) the combined training set surely includes inaccurate labels that might harm parser performance. Figure 2 (left) shows the increase in coverage achieved for in-domain and out-of-domain self-training data. The improvements induced by both methods are similar. This is quite surprising given that the Brown sections we used as self-training data contain science, fiction, humor, romance, mystery and adventure texts while the test section in these experiments, WSJ section 23, contains only news articles.

Figure 2 also compares recall (middle) and precision (right) for the different methods. For II there is a significant improvement in both precision and recall even though many more sentences are parsed. For IO, there is a large gain in recall and a much smaller loss in precision, yielding a substantial improvement in f-score (Figure 1).

F - score	This work - II	Becker unparsed	Becker entropy/unparsed	Hwa TE
80	**50%**	29.4%	30.6%	-5.7%

Table 3: Reduction of the number of manually annotated constituents needed for achieving f score value of 80 on section 23 of the WSJ. In all cases the seed and additional sentences selected to train the parser are taken from sections 02-21 of WSJ.

5.2 Out-of-domain seed data

In these two experiments we show that self-training is valuable for adapting parsers from one domain to another. Figure 3 compares out-of-domain seed data used with in-domain (OI) or out-of-domain (OO) self-training data against the baseline of training only with the out-of-domain seed data.

The left graph shows a significant improvement in f-score. In the middle and right graphs we examine the quality of the parses produced by the model by plotting recall and precision vs. seed size. Regarding precision, the difference between the three conditions is small relative to the f-score difference shown in the left graph. The improvement in the recall measure is much greater than the precision differences, and this is reflected in the f-score result. The gain in coverage achieved by both methods, which is not shown in the figure, is similar to that reported for the in-domain seed experiments. The left graph along with the increase in coverage show the power of self-training in parser adaptation when small seed datasets are used: not only do OO and OI parse many more sentences than the baseline, but their f-score values are consistently better.

To see how much manually annotated data can be saved by using out-of-domain seed, we train the parsing model with manually annotated data from the Brown training section, as described in Section 4. We assume that given a fixed number of training sentences the best performance of the parser without self-training will occur when these sentences are selected from the domain of the test section, the Brown corpus. We compare the amounts of manually annotated data needed to achieve certain f-score levels in this condition with the corresponding amounts of data needed by OI and OO. The results are summarized in Table 4. We compare to two baselines using in- and out-of-domain seed data without

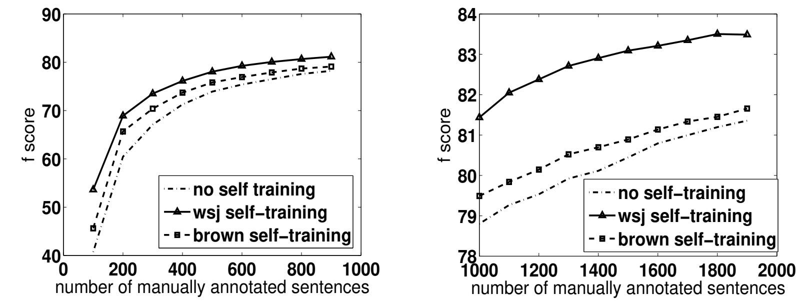

Figure 1: Number of seed sentences vs. f-score, for the two in-domain seed experiments: II (triangles) and IO (squares), and for the no self-training baseline. Self-training provides a substantial improvement.

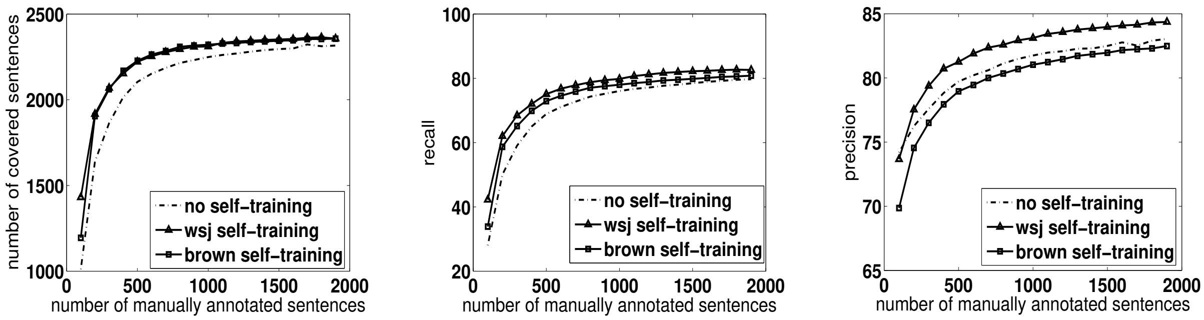

Figure 2: Number of seed sentences vs. coverage (left), recall (middle) and precision (right) for the two in-domain seed experiments: II (triangles) and IO (squares), and for the no self-training baseline.

any self-training. The second line (ID) serves as a reference to compute how much manual annotation of the test domain was saved, and the first line (OD) serves as a reference to show by how much self-training improves the out-of-domain baseline. The table stops at an f-score of 74 because that is the best that the baselines can do.

A significant reduction in annotation cost over the ID baseline is achieved where the seed size is between 100 and 1200. Improvement over the OD baseline is for the whole range of seed sizes. Both OO and OI achieve 20-33% reduction in manual annotation compared to the ID baseline and enhance the performance of the parser by as much as 42.9%.

The only previous work that adapts a parser trained on a small dataset between domains is that of (Steedman et al., 2003a), which used co-training (no self-training results were reported there or elsewhere). In order to compare with that work, we performed OI with seed taken from the Brown corpus and self-training and test taken from WSJ, which is the setup they use, obtaining a similar improvement to that reported there. However, co-training is a more complex method that requires an additional parser (LTAG in their case).

To further substantiate our results for the parser adaptation scenario, we used an additional corpus, Switchboard. Figure 4 shows the results of an OI experiment with WSJ seed and Switchboard self-training and test data. Although the domains of these two corpora are very different (more so than WSJ and Brown), self-training provides a substantial improvement.

We have also performed all four experiments with Brown and WSJ trading places. The results obtained were very similar to those reported here, and will not be detailed due to lack of space.

6 Analysis

In this section we try to better understand the benefit in using self-training with small seed datasets. We formulate the following criterion: the number of words in a test sentence that do not appear in the seed data ('unknown words') is a strong indicator

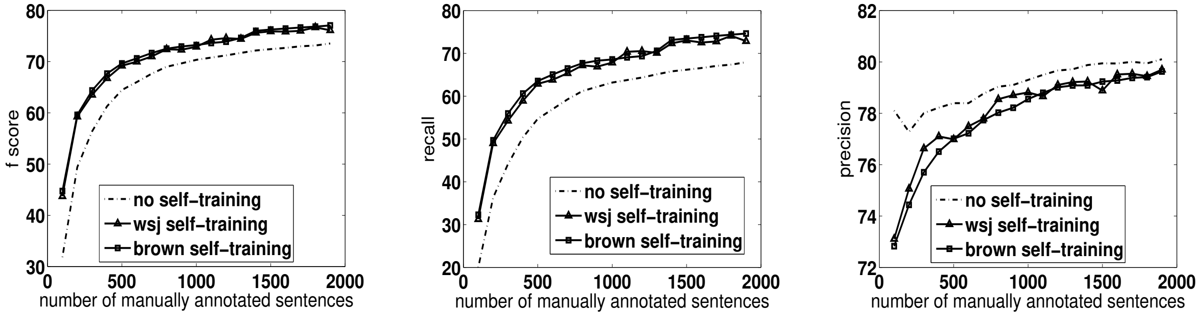

Figure 3: Number of seed sentences vs. f-score (left), recall (middle) and precision (right), for the two out-of-domain seed data experiments: OO (triangles) and OI (squares), and for the no self-training baseline.

f-sc.	66	68	70	72	74
OD	600	800	$1,000$	$1,400$	–
ID	600	700	800	$1,000$	$1,200$
OO	**400**	**500**	**600**	**800**	**1100**
	$33, 33$	$28.6, 37.5$	$33, 40$	$20, 42.9$	$8, -$
OI	**400**	**500**	**600**	**800**	$1,300$
	$33, 33$	$28.6, 37.5$	$33, 40$	$20, 42.9$	$-8, -$

Table 4: Number of manually annotated seed sentences needed for achieving certain f-score values. The first two lines show the out-of-domain and in-domain seed baselines. The reductions compared to the baselines is given as ID, OD.

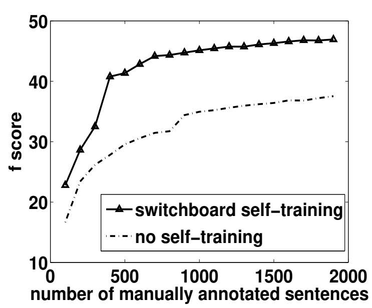

Figure 4: Number of seed sentences vs. f-score, for the OI experiment using WSJ seed data and SwitchBoard self-training and test data. In spite of the strong dissimilarity between the domains, self-training provides a substantial improvement.

to whether it is worthwhile to use small seed self-training. Figure 5 shows the number of unknown words in a sentence vs. the probability that the self-training model will parse a sentence no worse (upper curve) or better (lower curve) than the baseline model.

The upper curve shows that regardless of the

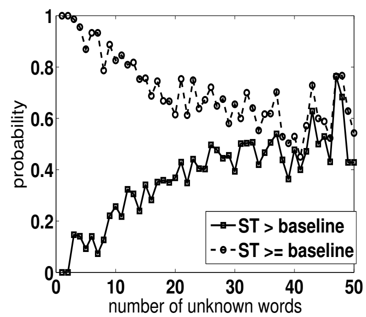

Figure 5: For sentences having the same number of unknown words, we show the probability that the self-training model parses a sentence from the set no worse (upper curve) or better (lower curve) than the baseline model.

number of unknown words in the sentence, there is more than 50% chance that the self-training model will not harm the result. This probability decreases from almost 1 for a very small number of unknown words to about 0.55 for 50 unknown words. The lower curve shows that when the number of unknown words increases, the probability that the self-training model will do better than the baseline model increases from almost 0 (for a very small number of unknown words) to about 0.55. Hence, the number of unknown words is an indication for the potential benefit (value on the lower curve) and risk (1 minus the value on the upper curve) in using the self-training model compared to using the baseline model. Unknown words were not identified in (McClosky et al., 2006a) as a useful predictor for the benefit of self-training.

We also identified a length effect similar to that studied by (McClosky et al., 2006a) for self-training (using a reranker and large seed, as detailed in Section 2). Due to space limitations we do not discuss it here.

7 Discussion

Self-training is usually not considered to be a valuable technique in improving the performance of generative statistical parsers, especially when the manually annotated seed sentence dataset is small. Indeed, in the II scenario, (Steedman et al., 2003a; McClosky et al., 2006a; Charniak, 1997) reported no improvement of the base parser for small (500 sentences, in the first paper) and large (40K sentences, in the last two papers) seed datasets respectively. In the II, OO, and OI scenarios, (McClosky et al, 2006a; 2006b) succeeded in improving the parser performance only when a reranker was used to reorder the 50-best list of the generative parser, with a seed size of 40K sentences. Bacchiani et al (2006) improved the parser performance in the OI scenario but their seed size was large (about 20K sentences).

In this paper we have shown that self-training can enhance the performance of generative parsers, without a reranker, in four in- and out-of-domain scenarios using a small seed dataset. For the II, IO and OO scenarios, we are the first to show improvement by self-training for generative parsers. We achieved a 50% (20-33%) reduction in annotation cost for the in-domain (out-of-domain) seed data scenarios. Previous work with small seed datasets considered only the II and OI scenarios. Our results for the former are better than any previous method, and our results for the latter (which are the first reported self-training results) are similar to previous results for co-training, a more complex method. We demonstrated our results using three corpora of varying degrees of domain difference.

A direction for future research is combining self-training data from various domains to enhance parser adaptation.

Acknowledgement. We would like to thank Dan Roth for his constructive comments on this paper.

References

Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat, 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Markus Becker and Miles Osborne, 2005. A two-stage method for active learning of statistical grammars. *IJCAI '05.*

Daniel Bikel, 2004. *Code developed at University of Pennsylvania*. http://www.cis.upenn.edu.bikel.

Avrim Blum and Tom M. Mitchell, 1998. Combining labeled and unlabeled data with co-training. *COLT '98.*

Eugene Charniak, 1997. Statistical parsing with a context-free grammar and word statistics. *AAAI '97.*

Eugene Charniak, 2000. A maximum-entropy-inspired parser. *ANLP '00.*

Eugene Charniak and Mark Johnson, 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. *ACL '05.*

Stephen Clark, James Curran, and Miles Osborne, 2003. Bootstrapping pos taggers using unlabelled data. *CoNLL '03.*

David A. Cohn, Les Atlas, and Richard E. 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.

Rebecca Hwa, Miles Osborne, Anoop Sarkar and Mark Steedman, 2003. Corrected co-training for statistical parsers. In *ICML '03, Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining.*

Rebecca Hwa, 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

Terry Koo and Michael Collins, 2005. Hidden-variable models for discriminative reranking. *EMNLP '05.*

David McClosky, Eugene Charniak, and Mark Johnson, 2006a. Effective self-training for parsing. *HLT-NAACL '06.*

David McClosky, Eugene Charniak, and Mark Johnson, 2006b. Reranking and self-training for parser adaptation. *ACL-COLING '06.*

Mark Steedman, Anoop Sarkar, Miles Osborne, Rebecca Hwa, Stephen Clark, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim, 2003a. Bootstrapping statistical parsers from small datasets. *EACL '03.*

Mark Steedman, Rebecca Hwa, Stephen Clark, Miles Osborne, Anoop Sarkar, Julia Hockenmaier, Paul Ruhlen,Steven Baker, and Jeremiah Crim, 2003b. Example selection for bootstrapping statistical parsers. *NAACL '03.*