Reranking and Self-Training for Parser Adaptation

David McClosky, Eugene Charniak, and Mark Johnson
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI 02912
{dmcc|ec|mj}@cs.brown.edu

Abstract

Statistical parsers trained and tested on the Penn Wall Street Journal (WSJ) treebank have shown vast improvements over the last 10 years. Much of this improvement, however, is based upon an ever-increasing number of features to be trained on (typically) the WSJ treebank data. This has led to concern that such parsers may be too finely tuned to this corpus at the expense of portability to other genres. Such worries have merit. The standard "Charniak parser" checks in at a labeled precision-recall f-measure of 89.7% on the Penn WSJ test set, but only 82.9% on the test set from the Brown treebank corpus.

This paper should allay these fears. In particular, we show that the reranking parser described in Charniak and Johnson (2005) improves performance of the parser on Brown to 85.2%. Furthermore, use of the self-training techniques described in (McClosky et al., 2006) raise this to 87.8% (an error reduction of 28%) again without any use of labeled Brown data. This is remarkable since training the parser and reranker on labeled Brown data achieves only 88.4%.

1 Introduction

Modern statistical parsers require treebanks to train their parameters, but their performance declines when one parses genres more distant from the training data's domain. Furthermore, the treebanks required to train said parsers are expensive and difficult to produce.

Naturally, one of the goals of statistical parsing is to produce a broad-coverage parser which is relatively insensitive to textual domain. But the lack of corpora has led to a situation where much of the current work on parsing is performed on a single domain using training data from that domain — the Wall Street Journal (WSJ) section of the Penn Treebank (Marcus et al., 1993). Given the aforementioned costs, it is unlikely that many significant treebanks will be created for new genres. Thus, *parser adaptation* attempts to leverage existing labeled data from one domain and create a parser capable of parsing a different domain.

Unfortunately, the state of the art in parser portability (i.e. using a parser trained on one domain to parse a different domain) is not good. The "Charniak parser" has a labeled precision-recall f-measure of 89.7% on WSJ but a lowly 82.9% on the test set from the Brown corpus treebank. Furthermore, the treebanked Brown data is mostly general non-fiction and much closer to WSJ than, e.g., medical corpora would be. Thus, most work on parser adaptation resorts to using some labeled in-domain data to fortify the larger quantity of out-of-domain data.

In this paper, we present some encouraging results on parser adaptation without any in-domain data. (Though we also present results with in-domain data as a reference point.) In particular we note the effects of two comparatively recent techniques for parser improvement.

The first of these, *parse-reranking* (Collins, 2000; Charniak and Johnson, 2005) starts with a "standard" generative parser, but uses it to generate the n-best parses rather than a single parse. Then a reranking phase uses more detailed features, features which would (mostly) be impossible to incorporate in the initial phase, to reorder

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 337–344,
Sydney, July 2006. ©2006 Association for Computational Linguistics

the list and pick a possibly different best parse. At first blush one might think that gathering even more fine-grained features from a WSJ treebank would not help adaptation. However, we find that reranking improves the parsers performance from 82.9% to 85.2%.

The second technique is *self-training* — parsing unlabeled data and adding it to the training corpus. Recent work, (McClosky et al., 2006), has shown that adding many millions of words of machine parsed and reranked LA Times articles does, in fact, improve performance of the parser on the closely related WSJ data. Here we show that it also helps the father-afield Brown data. Adding it improves performance yet-again, this time from 85.2% to 87.8%, for a net error reduction of 28%. It is interesting to compare this to our results for a completely Brown trained system (i.e. one in which the first-phase parser is trained on just Brown training data, and the second-phase reranker is trained on Brown 50-best lists). This system performs at a 88.4% level — only slightly higher than that achieved by our system with only WSJ data.

2 Related Work

Work in parser adaptation is premised on the assumption that one wants a single parser that can handle a wide variety of domains. While this is the goal of the majority of parsing researchers, it is not quite universal. Sekine (1997) observes that for parsing a specific domain, data from that domain is most beneficial, followed by data from the same class, data from a different class, and data from a different domain. He also notes that different domains have very different structures by looking at frequent grammar productions. For these reasons he takes the position that we should, instead, simply create treebanks for a large number of domains. While this is a coherent position, it is far from the majority view.

There are many different approaches to parser adaptation. Steedman et al. (2003) apply co-training to parser adaptation and find that co-training can work across domains. The need to parse biomedical literature inspires (Clegg and Shepherd, 2005; Lease and Charniak, 2005). Clegg and Shepherd (2005) provide an extensive side-by-side performance analysis of several modern statistical parsers when faced with such data. They find that techniques which combine different parsers such as voting schemes and parse selection can improve performance on biomedical data. Lease and Charniak (2005) use the Charniak parser for biomedical data and find that the use of out-of-domain trees and in-domain vocabulary information can considerably improve performance.

Training	Testing	f-measure	
		Gildea	Bacchiani
WSJ	WSJ	86.4	87.0
WSJ	Brown	80.6	81.1
Brown	Brown	84.0	84.7
WSJ+Brown	Brown	84.3	85.6

Table 1: Gildea and Bacchiani results on WSJ and Brown test corpora using different WSJ and Brown training sets. Gildea evaluates on sentences of length ≤ 40, Bacchiani on all sentences.

However, the work which is most directly comparable to ours is that of (Ratnaparkhi, 1999; Hwa, 1999; Gildea, 2001; Bacchiani et al., 2006). All of these papers look at what happens to modern WSJ-trained statistical parsers (Ratnaparkhi's, Collins', Gildea's and Roark's, respectively) as training data varies in size or usefulness (because we are testing on something other than WSJ). We concentrate particularly on the work of (Gildea, 2001; Bacchiani et al., 2006) as they provide results which are directly comparable to those presented in this paper.

Looking at Table 1, the first line shows us the standard training and testing on WSJ — both parsers perform in the 86-87% range. The next line shows what happens when parsing Brown using a WSJ-trained parser. As with the Charniak parser, both parsers take an approximately 6% hit.

It is at this point that our work deviates from these two papers. Lacking alternatives, both (Gildea, 2001) and (Bacchiani et al., 2006) give up on adapting a pure WSJ trained system, instead looking at the issue of how much of an improvement one gets over a pure Brown system by adding WSJ data (as seen in the last two lines of Table 1). Both systems use a "model-merging" (Bacchiani et al., 2006) approach. The different corpora are, in effect, concatenated together. However, (Bacchiani et al., 2006) achieve a larger gain by weighting the in-domain (Brown) data more heavily than the out-of-domain WSJ data. One can imagine, for instance, five copies of the Brown data concatenated with just one copy of WSJ data.

3 Corpora

We primarily use three corpora in this paper. Self-training requires labeled and unlabeled data. We assume that these sets of data must be in similar domains (e.g. news articles) though the effectiveness of self-training across domains is currently an open question. Thus, we have labeled (WSJ) and unlabeled (NANC) out-of-domain data and labeled in-domain data (BROWN). Unfortunately, lacking a corresponding corpus to NANC for BROWN, we cannot perform the opposite scenario and adapt BROWN to WSJ.

3.1 Brown

The BROWN corpus (Francis and Kučera, 1979) consists of many different genres of text, intended to approximate a "balanced" corpus. While the full corpus consists of fiction and nonfiction domains, the sections that have been annotated in Treebank II bracketing are primarily those containing fiction. Examples of the sections annotated include science fiction, humor, romance, mystery, adventure, and "popular lore." We use the same divisions as Bacchiani et al. (2006), who base their divisions on Gildea (2001). Each division of the corpus consists of sentences from all available genres. The training division consists of approximately 80% of the data, while held-out development and testing divisions each make up 10% of the data. The treebanked sections contain approximately 25,000 sentences (458,000 words).

3.2 Wall Street Journal

Our out-of-domain data is the Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus et al., 1993) which consists of about 40,000 sentences (one million words) annotated with syntactic information. We use the standard divisions: Sections 2 through 21 are used for training, section 24 for held-out development, and section 23 for final testing.

3.3 North American News Corpus

In addition to labeled news data, we make use of a large quantity of unlabeled news data. The unlabeled data is the North American News Corpus, NANC (Graff, 1995), which is approximately 24 million unlabeled sentences from various news sources. NANC contains no syntactic information and sentence boundaries are induced by a simple discriminative model. We also perform some basic cleanups on NANC to ease parsing. NANC contains news articles from various news sources including the Wall Street Journal, though for this paper, we only use articles from the LA Times portion.

To use the data from NANC, we use *self-training* (McClosky et al., 2006). First, we take a WSJ trained reranking parser (i.e. both the parser and reranker are built from WSJ training data) and parse the sentences from NANC with the 50-best (Charniak and Johnson, 2005) parser. Next, the 50-best parses are reordered by the reranker. Finally, the 1-best parses after reranking are combined with the WSJ training set to retrain the first-stage parser.[1] McClosky et al. (2006) find that the self-trained models help considerably when parsing WSJ.

4 Experiments

We use the Charniak and Johnson (2005) reranking parser in our experiments. Unless mentioned otherwise, we use the WSJ-trained reranker (as opposed to a BROWN-trained reranker). To evaluate, we report bracketing f-scores.[2] Parser f-scores reported are for sentences up to 100 words long, while reranking parser f-scores are over all sentences. For simplicity and ease of comparison, most of our evaluations are performed on the development section of BROWN.

4.1 Adapting self-training

Our first experiment examines the performance of the self-trained parsers. While the parsers are created entirely from labeled WSJ data and unlabeled NANC data, they perform extremely well on BROWN development (Table 2). The trends are the same as in (McClosky et al., 2006): Adding NANC data improves parsing performance on BROWN development considerably, improving the f-score from 83.9% to 86.4%. As more NANC data is added, the f-score appears to approach an asymptote. The NANC data appears to help reduce data sparsity and fill in some of the gaps in the WSJ model. Additionally, the reranker provides further benefit and adds an absolute 1-2% to the f-score. The improvements appear to be orthogonal, as our best performance is reached when we use the reranker and add 2,500k self-trained sentences from NANC.

[1] We trained a new reranker from this data as well, but it does not seem to get significantly different performance.

[2] The harmonic mean of labeled precision (P) and labeled recall (R), i.e. $f = \frac{2 \times P \times R}{P + R}$

Sentences added	Parser	Reranking Parser
Baseline BROWN	86.4	87.4
Baseline WSJ	83.9	85.8
WSJ+50k	84.8	86.6
WSJ+250k	85.7	87.2
WSJ+500k	86.0	87.3
WSJ+750k	86.1	87.5
WSJ+1,000k	86.2	87.3
WSJ+1,500k	86.2	87.6
WSJ+2,000k	86.1	87.7
WSJ+2,500k	86.4	87.7

Table 2: Effects of adding NANC sentences to WSJ training data on parsing performance. f-scores for the parser with and without the WSJ reranker are shown when evaluating on BROWN development. For this experiment, we use the WSJ-trained reranker.

The results are even more surprising when we compare against a parser[3] trained on the labeled training section of the BROWN corpus, with parameters tuned against its held-out section. Despite seeing no in-domain data, the WSJ based parser is able to match the BROWN based parser.

For the remainder of this paper, we will refer to the model trained on WSJ+2,500k sentences of NANC as our "best WSJ+NANC" model. We also note that this "best" parser is different from the "best" parser for parsing WSJ, which was trained on WSJ with a relative weight[4] of 5 and 1,750k sentences from NANC. For parsing BROWN, the difference between these two parsers is not large, though.

Increasing the relative weight of WSJ sentences versus NANC sentences when testing on BROWN development does not appear to have a significant effect. While (McClosky et al., 2006) showed that this technique was effective when testing on WSJ, the true distribution was closer to WSJ so it made sense to emphasize it.

4.2 Incorporating In-Domain Data

Up to this point, we have only considered the situation where we have no in-domain data. We now explore different ways of making use of labeled and unlabeled in-domain data.

Bacchiani et al. (2006) applies self-training to parser adaptation to utilize unlabeled in-domain data. The authors find that it helps quite a bit when adapting from BROWN to WSJ. They use a parser trained from the BROWN train set to parse WSJ and add the parsed WSJ sentences to their training set. We perform a similar experiment, using our WSJ-trained reranking parser to parse BROWN train and testing on BROWN development. We achieved a boost from 84.8% to 85.6% when we added the parsed BROWN sentences to our training. Adding in 1,000k sentences from NANC as well, we saw a further increase to 86.3%. However, the technique does not seem as effective in our case. While the self-trained BROWN data helps the parser, it adversely affects the performance of the reranking parser. When self-trained BROWN data is added to WSJ training, the reranking parser's performance drops from 86.6% to 86.1%. We see a similar degradation as NANC data is added to the training set as well. We are not yet able to explain this unusual behavior.

We now turn to the scenario where we have some labeled in-domain data. The most obvious way to incorporate labeled in-domain data is to combine it with the labeled out-of-domain data. We have already seen the results (Gildea, 2001) and (Bacchiani et al., 2006) achieve in Table 1.

We explore various combinations of BROWN, WSJ, and NANC corpora. Because we are mainly interested in exploring techniques with self-trained models rather than optimizing performance, we only consider weighting each corpus with a relative weight of one for this paper. The models generated are tuned on section 24 from WSJ. The results are summarized in Table 3.

While both WSJ and BROWN models benefit from a small amount of NANC data, adding more than 250k NANC sentences to the BROWN or combined models causes their performance to drop. This is not surprising, though, since adding "too much" NANC overwhelms the more accurate BROWN or WSJ counts. By weighting the counts from each corpus appropriately, this problem can be avoided.

Another way to incorporate labeled data is to tune the parser back-off parameters on it. Bacchiani et al. (2006) report that tuning on held-out BROWN data gives a large improvement over tun-

[3] In this case, only the parser is trained on BROWN. In section 4.3, we compare against a fully BROWN-trained reranking parser as well.

[4] A relative weight of n is equivalent to using n copies of the corpus, i.e. an event that occurred x times in the corpus would occur $x \times n$ times in the weighted corpus. Thus, larger corpora will tend to dominate smaller corpora of the same relative weight in terms of event counts.

ing on WSJ data. The improvement is mostly (but not entirely) in precision. We do not see the same improvement (Figure 1) but this is likely due to differences in the parsers. However, we do see a similar improvement for parsing accuracy once NANC data has been added. The reranking parser generally sees an improvement, but it does not appear to be significant.

4.3 Reranker Portability

We have shown that the WSJ-trained reranker is actually quite portable to the BROWN fiction domain. This is surprising given the large number of features (over a million in the case of the WSJ reranker) tuned to adjust for errors made in the 50-best lists by the first-stage parser. It would seem the corrections memorized by the reranker are not as domain-specific as we might expect.

As further evidence, we present the results of applying the WSJ model to the Switchboard corpus — a domain much less similar to WSJ than BROWN. In Table 4, we see that while the parser's performance is low, self-training and reranking provide orthogonal benefits. The improvements represent a 12% error reduction with no additional in-domain data. Naturally, in-domain data and speech-specific handling (e.g. disfluency modeling) would probably help dramatically as well.

Finally, to compare against a model fully trained on BROWN data, we created a BROWN reranker. We parsed the BROWN training set with 20-fold cross-validation, selected features that occurred 5 times or more in the training set, and fed the 50-best lists from the parser to a numerical optimizer to estimate feature weights. The resulting reranker model had approximately 700,000 features, which is about half as many as the WSJ trained reranker. This may be due to the smaller size of the BROWN training set or because the feature schemas for the reranker were developed on WSJ data. As seen in Table 5, the BROWN reranker is not a significant improvement over the WSJ reranker for parsing BROWN data.

5 Analysis

We perform several types of analysis to measure some of the differences and similarities between the BROWN-trained and WSJ-trained reranking parsers. While the two parsers agree on a large number of parse brackets (Section 5.2), there are categorical differences between them (as seen in Section 5.3).

Parser model	Parser f-score	Reranker f-score
WSJ	74.0	75.9
WSJ+NANC	75.6	77.0

Table 4: Parser and reranking parser performance on the SWITCHBOARD development corpus. In this case, WSJ+NANC is a model created from WSJ and 1,750k sentences from NANC.

Model	1-best	10-best	25-best	50-best
WSJ	82.6	88.9	90.7	91.9
WSJ+NANC	86.4	92.1	93.5	94.3
BROWN	86.3	92.0	93.3	94.2

Table 6: Oracle f-scores of top n parses produced by baseline WSJ parser, a combined WSJ and NANC parser, and a baseline BROWN parser.

5.1 Oracle Scores

Table 6 shows the f-scores of an "oracle reranker" — i.e. one which would always choose the parse with the highest f-score in the n-best list. While the WSJ parser has relatively low f-scores, adding NANC data results in a parser with comparable oracle scores as the parser trained from BROWN training. Thus, the WSJ+NANC model has better oracle rates than the WSJ model (McClosky et al., 2006) for both the WSJ and BROWN domains.

5.2 Parser Agreement

In this section, we compare the output of the WSJ+NANC-trained and BROWN-trained reranking parsers. We use *evalb* to calculate how similar the two sets of output are on a bracket level. Table 7 shows various statistics. The two parsers achieved an 88.0% f-score between them. Additionally, the two parsers agreed on all brackets almost half the time. The part of speech tagging agreement is fairly high as well. Considering they were created from different corpora, this seems like a high level of agreement.

5.3 Statistical Analysis

We conducted randomization tests for the significance of the difference in corpus f-score, based on the randomization version of the paired sample t-test described by Cohen (1995). The null hypothesis is that the two parsers being compared are in fact behaving identically, so permuting or swapping the parse trees produced by the parsers for

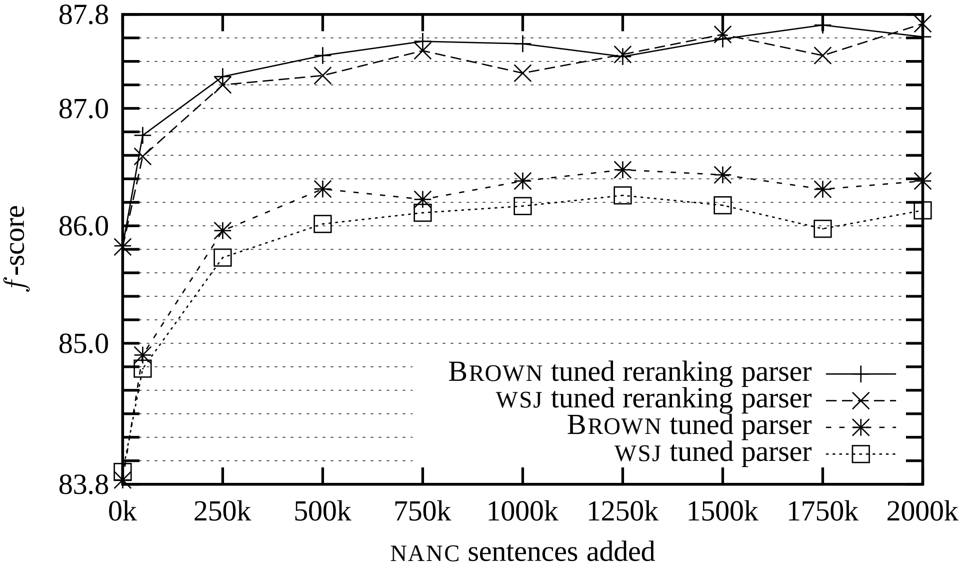

Figure 1: Precision and recall *f*-scores when testing on BROWN development as a function of the number of NANC sentences added under four test conditions. "BROWN tuned" indicates that BROWN training data was used to tune the parameters (since the normal held-out section was being used for testing). For "WSJ tuned," we tuned the parameters from section 24 of WSJ. Tuning on BROWN helps the parser, but not for the reranking parser.

Parser model	Parser alone	Reranking parser
WSJ alone	83.9	85.8
WSJ+2,500k NANC	86.4	87.7
BROWN alone	86.3	87.4
BROWN+50k NANC	86.8	88.0
BROWN+250k NANC	86.8	88.1
BROWN+500k NANC	86.7	87.8
WSJ+BROWN	86.5	88.1
WSJ+BROWN+50k NANC	86.8	88.1
WSJ+BROWN+250k NANC	86.8	88.1
WSJ+BROWN+500k NANC	86.6	87.7

Table 3: *f*-scores from various combinations of WSJ, NANC, and BROWN corpora on BROWN development. The reranking parser used the WSJ-trained reranker model. The BROWN parsing model is naturally better than the WSJ model for this task, but combining the two training corpora results in a better model (as in Gildea (2001)). Adding small amounts of NANC further improves the models.

Parser model	Parser alone	WSJ-reranker	BROWN-reranker
WSJ	82.9	85.2	85.2
WSJ+NANC	87.1	87.8	87.9
BROWN	86.7	88.2	88.4

Table 5: Performance of various combinations of parser and reranker models when evaluated on BROWN test. The WSJ+NANC parser with the WSJ reranker comes close to the BROWN-trained reranking parser. The BROWN reranker provides only a small improvement over its WSJ counterpart, which is not statistically significant.

Bracketing agreement f-score	88.03%
Complete match	44.92%
Average crossing brackets	0.94
POS Tagging agreement	94.85%

Table 7: Agreement between the WSJ+NANC parser with the WSJ reranker and the BROWN parser with the BROWN reranker. Complete match is how often the two reranking parsers returned the exact same parse.

the same test sentence should not affect the corpus f-scores. By estimating the proportion of permutations that result in an absolute difference in corpus f-scores at least as great as that observed in the actual output, we obtain a distribution-free estimate of significance that is robust against parser and evaluator failures. The results of this test are shown in Table 8. The table shows that the BROWN reranker is not significantly different from the WSJ reranker.

In order to better understand the difference between the reranking parser trained on Brown and the WSJ+NANC/WSJ reranking parser (a reranking parser with the first-stage trained on WSJ+NANC and the second-stage trained on WSJ) on Brown data, we constructed a logistic regression model of the difference between the two parsers' f-scores on the development data using the R statistical package[5]. Of the 2,078 sentences in the development data, 29 sentences were discarded because *evalb* failed to evaluate at least one of the parses.[6] A Wilcoxon signed rank test on the remaining 2,049 paired sentence level f-scores was significant at $p = 0.0003$. Of these 2,049 sentences, there were 983 parse pairs with the same sentence-level f-score. Of the 1,066 sentences for which the parsers produced parses with different f-scores, there were 580 sentences for which the BROWN/BROWN parser produced a parse with a higher sentence-level f-score and 486 sentences for which the WSJ+NANC/WSJ parser produce a parse with a higher f-score. We constructed a generalized linear model with a binomial link with BROWN/BROWN f-score $>$ WSJ+NANC/WSJ f-score as the predicted variable, and sentence length, the number of prepositions (IN), the number of conjunctions (CC) and Brown subcorpus ID as explanatory variables. Model selection (using the "step" procedure) discarded all but the IN and Brown ID explanatory variables. The final estimated model is shown in Table 9. It shows that the WSJ+NANC/WSJ parser becomes more likely to have a higher f-score than the BROWN/BROWN parser as the number of prepositions in the sentence increases, and that the BROWN/BROWN parser is more likely to have a higher f-score on Brown sections K, N, P, G and L (these are the general fiction, adventure and western fiction, romance and love story, letters and memories, and mystery sections of the Brown corpus, respectively). The three sections of BROWN not in this list are F, M, and R (popular lore, science fiction, and humor).

Feature	Estimate	z-value	$\Pr(>\lvert z\rvert)$	
(Intercept)	0.054	0.3	0.77	
IN	-0.134	-4.4	8.4e-06	***
ID=G	0.584	2.5	0.011	*
ID=K	0.697	2.9	0.003	**
ID=L	0.552	2.3	0.021	*
ID=M	0.376	0.9	0.33	
ID=N	0.642	2.7	0.0055	**
ID=P	0.624	2.7	0.0069	**
ID=R	0.040	0.1	0.90	

Table 9: The logistic model of BROWN/BROWN f-score $>$ WSJ+NANC/WSJ f-score identified by model selection. The feature IN is the number prepositions in the sentence, while ID identifies the Brown subcorpus that the sentence comes from. Stars indicate significance level.

6 Conclusions and Future Work

We have demonstrated that rerankers and self-trained models can work well across domains. Models self-trained on WSJ appear to be better parsing models in general, the benefits of which are not limited to the WSJ domain. The WSJ-trained reranker using out-of-domain LA Times parses (produced by the WSJ-trained reranker) achieves a labeled precision-recall f-measure of 87.8% on Brown data, nearly equal to the performance one achieves by using a purely Brown trained parser-reranker. The 87.8% f-score on Brown represents a 24% error reduction on the corpus.

Of course, as corpora differences go, Brown is relatively close to WSJ. While we also find that our

[5] http://www.r-project.org

[6] This occurs when an apostrophe is analyzed as a possessive marker in the gold tree and a punctuation symbol in the parse tree, or vice versa.

	WSJ+NANC/WSJ	BROWN/WSJ	BROWN/BROWN
WSJ/WSJ	0.025 (0)	0.030 (0)	0.031 (0)
WSJ+NANC/WSJ		0.004 (0.1)	0.006 (0.025)
BROWN/WSJ			0.002 (0.27)

Table 8: The difference in corpus f-score between the various reranking parsers, and the significance of the difference in parentheses as estimated by a randomization test with 10^6 samples. "x/y" indicates that the first-stage parser was trained on data set x and the second-stage reranker was trained on data set y.

"best" WSJ-parser-reranker improves performance on the Switchboard corpus, it starts from a much lower base (74.0%), and achieves a much less significant improvement (3% absolute, 11% error reduction). Bridging these larger gaps is still for the future.

One intriguing idea is what we call "self-trained bridging-corpora." We have not yet experimented with medical text but we expect that the "best" WSJ+NANC parser will not perform very well. However, suppose one does self-training on a biology textbook instead of the LA Times. One might hope that such a text will split the difference between more "normal" newspaper articles and the specialized medical text. Thus, a self-trained parser based upon such text might do much better than our standard "best." This is, of course, highly speculative.

Acknowledgments

This work was supported by NSF grants LIS9720368, and IIS0095940, and DARPA GALE contract HR0011-06-2-0001. We would like to thank the BLLIP team for their comments.

References

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

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *Proc. of the 2005 Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 173–180.

Andrew B. Clegg and Adrian Shepherd. 2005. Evaluating and integrating treebank parsers on a biomedical corpus. In *Proceedings of the ACL Workshop on Software*.

Paul R. Cohen. 1995. *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML 2000)*, pages 175–182, Stanford, California.

W. Nelson Francis and Henry Kučera. 1979. *Manual of Information to accompany a Standard Corpus of Present-day Edited American English*, for use with Digital Computers. Brown University, Providence, Rhode Island.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202.

David Graff. 1995. *North American News Text Corpus*. Linguistic Data Consortium. LDC95T21.

Rebecca Hwa. 1999. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 72–80, University of Maryland.

Matthew Lease and Eugene Charniak. 2005. Parsing biomedical literature. In *Second International Joint Conference on Natural Language Processing (IJCNLP'05)*.

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of HLT-NAACL 2006*.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175.

Satoshi Sekine. 1997. The domain dependence of parsing. In *Proc. Applied Natural Language Processing (ANLP)*, pages 96–102.

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 *Proc. of European ACL (EACL)*, pages 331–338.