Domain Adaptation for Parsing

Eric Baucom
Indiana University
Bloomington, IN, USA
eabaucom@indiana.edu

Levi King
Indiana University
Bloomington, IN, USA
leviking@indiana.edu

Sandra Kübler
Indiana University
Bloomington, IN, USA
skuebler@indiana.edu

Abstract

We compare two different methods in domain adaptation applied to constituent parsing: parser combination and co-training, each used to transfer information from the source domain of news to the target domain of natural dialogs, in a setting without annotated data. Both methods outperform the baselines and reach similar results. Parser combination profits most from the large amounts of training data combined with a robust probability model. Co-training, in contrast, relies on a small set of higher quality data.

1 Introduction

Research on parsing has mostly concentrated on parsing the Penn Treebank (Marcus et al., 1993). As a consequence, most parsers have probability models that are optimized for the syntactic annotations in this treebank and more generally for the language in the treebank. This means that a parser trained on the Penn Treebank will show a severe degradation in performance when used for parsing data from another domain (McClosky et al., 2010). More recently, research has started on adapting parsers to new domains so that the degradation in parsing is minimized. One of the first venues at which domain adaptation was targeted was the 2007 CoNLL shared task on dependency parsing (Nivre et al., 2007).

One of the challenges in domain adaptation for parsing is the lack of annotated data in the target domain. Research has covered a range of different approaches, all geared towards providing automatically labeled data in the target domain to add as training data. Approaches include ensembles of parsers, self-training, and methods for selecting high quality sentences to reduce the noise (see section 2 for details). The most promising approach at present is an approach by McClosky et al. (2010), which automatically selects a domain that is the most similar to the target domain.

In our current work, we investigate domain adaptation for constituent parsing, in a setting where no labeled data in the target domain is available. More specifically, we compare two different approaches: One approach is based on an ensemble of parsers, the other one uses co-training with two different parsers. Both approaches reach moderate improvements over the baseline, and we are interested in seeing the advantages and disadvantages of those two promising methods. The source domain for our experiments is the Penn Treebank; the target domain consists of spontaneous dialogs based on cooperative tasks involving navigation on a map or in a search environment. For the unlabeled target domain data, we use the Edinburgh Map Task (HCRC) corpus (Thompson et al., 1996), and the Indiana Cooperative Remote Search Task (CReST) corpus (Eberhard et al., 2010) as test set.

The remainder of the paper is structured as follows: In section 2, we discuss related work. Section 3 introduces the two methods that we will compare, and section 4 describes the experimental setup. In section 5, we first discuss the results of the individual approaches, and then attempt a comparison and an error analysis. In section 6, we conclude and describe future work.

2 Related Work

Domain adaptation can be divided into two different scenarios: one where a small set of annotated data from the target domain is available, and one where no annotated target data is available. Early work on domain adaptation for parsing shows that not having target domain data makes the task extremely challenging: In the CoNLL 2007 shared task on dependency parsing (Nivre et al., 2007), no team submitting results for the out-of-domain setting improved much over the baseline. Dredze et al. (2007), for example, presented three approaches to domain adaptation: modifications to the feature set, using a parser ensemble, and target focused learning, but they reached the best results by using all the available data. The best performing system (Sagae and Tsujii, 2007) used a combination of two different models of an LR parser and then selected identically parsed target sentences to add to the training set of the final parser. This approach outperformed the baseline of Dredze et al. (2007) by approximately 1%.

McClosky et al. (2006) use self-training in combination with a PCFG parser and reranking. They train the parser and reranker on the Penn Treebank, then parse and rerank a small set of target domain data. They reach an error reduction of 28% in the target domain. However, Sagae (2010) shows that while the reranking approach by McClosky et al. (2006) reaches higher F-scores than a self-training approach without reranking, the latter actually performs better in a semantic role labeling task.

Reichart and Rappoport (2007), in contrast, use a small annotated data set in the target domain for self-training without reranking. I.e., they train the parser on their small target domain data set and then perform self-training on more unlabeled data. They evaluate their parser in terms of annotation cost, and they show a 50% reduction in annotation cost.

Chen et al. (2008) work on domain adaptation without labeled target data: They parse the target data with a dependency parser. But rather than using the full parses as additional training data, they only add short-distance dependencies, which can be parsed more reliably. They gain approx. 1% over adding all sentences in Chinese. Kawahara and Uchimoto (2008) use a similar approach: They train a classifier to recognize reliably parsed sentences to add to the training set. This method outperforms the source domain baseline as well as all CoNLL 2007 systems by approx. 1%.

Finkel and Manning (2009) extend the work by Daume III (2007), who investigated a method for selecting general features that hold across domains. Finkel and Manning (2009) apply this method to dependency parsing, by using a hierarchical Bayesian model. They show an improvement of their approach over training on data from all domains in 4 out of 6 domains.

McClosky et al. (2010) investigate the automatic selection of source domains that are useful for parsing a target domain. Thus, the parser can adapt per document to a new target domain. They use different similarity metrics to determine the similarity of different source domains to the target domain and feed those into a regression model. They show that their model outperforms self-training, a uniform model as well as the best single domain for training selected by an oracle.

Miceli Barone and Attardi (2012) perform domain adaptation for dependency parsing using unannotated data. They integrate a transductive SVM as classifier, which can handle labeled and unlabeled examples as training data, into a shift-reduce dependency parser. They also reach an improvement in the area of 1% on Italian.

This overview shows that most work concentrates on domain adaptation when no annotated data in the target domain is available or when the target domain is unknown. Our work also focuses on a scenario where there is only unlabeled target domain data available. We compare co-training, a method that has not been used successfully for domain adaptation in parsing before, and a simpler approach based on an ensemble of three different parsers.

3 Domain Adaptation Methods

3.1 Parser Combination

A simple way of creating additional, labeled training data in a new domain is to use an ensemble of parsers and then select the sentences on which the parsers agree. This parser combination method takes advantage of the different biases built into different parsing algorithms; agreement between parsers should translate into a greater likelihood that the agreed upon parse will be correct.

In practice, the ensemble of parsers is trained on an available annotated data set in the source domain, i.e., the Penn Treebank (PTB) for parsing. They are then used to parse a corpus of unannotated data in the target domain. The sentences from the unannotated target domain on which the parsers agree are added to the original source domain gold-standard annotated data, and one (or more) parser is retrained on the resulting union.

Originally, this method was used by van Halteren et al. (2001) to improve part of speech taggers. Following Sagae and Tsujii (2007) and (Chen et al., 2008), we adapt the approach to the task of parsing by retraining with agreed upon parses from only a part of the ensemble as well as with partial trees. This will lead to more training data, though potentially of a lower quality.

3.2 Co-Training

Co-training, as proposed by Blum and Mitchell (1998), is a semi-supervised machine learning approach that uses two different "views" of the data to train two specialized classifiers, which provide additional training data for each other. In co-training for domain adaptation of parsers, we follow Goldman and Zhou (2000) in assuming two different parsers rather than two different feature sets. In other words, the different views come from two parsers built on different parsing algorithms, i.e., with different biases. The source-trained parsers are used to parse the unlabeled target data, providing a confidence score with each parse. The parsers each parse sentences from a pool of m randomly selected sentences from the set of unlabeled target domain data. The output of each parser is ranked by confidence scores, and the n-best parsed sentences from each parser are added to the original training data for the next cycle. Then the set of sentences is replenished from the unlabeled data set. This process is repeated until no further improvement on the development set is observed. Because the parsers have different algorithms and, theoretically, different strengths, each should be able to learn from highly-confident training data provided by the other.

4 Experimental Setup

4.1 Data Sets

We use the following data sets: The Penn Treebank (PTB) (Marcus et al., 1993) serves as the training set from the source domain. The PTB training files were modified to remove any grammatical functions not present in our target domain (see below). All experiments use either sections 2-11 (as in the 2007 CoNLL shared task on domain adaptation), or sections 2-21 (the standard training set for parsing).

Our target domain is dialog text taken from cooperative map tasks. The test corpus consists of Cooperative Remote Search Task (CReST) dialogs, in which a *searcher* collects and deposits items throughout a search location (a series of connected offices) at the guidance of a *director*, who has a map of the location and communicates instructions remotely by mobile telephone. The original CReST corpus contains a small number of novel tags to handle phenomena that are common in dialog data but not in newspaper text, such as imperative verbs. These tags were converted to their closest equivalents in the PTB tagset. The syntactic annotation of the CReST corpus includes constituent and dependency annotations. We use the constituent annotation, which follows the PTB annotation (Santorini, 1991). In contrast to the PTB, the CReST annotations use a subset of the grammatical functions from the Penn Treebank: subject, predicate, location, direction, and temporal modifications. For our experiments, 5 dialogs (1 137 sentences) of the CReST corpus were reserved for development, and 18 dialogs (4 518 sentences) were used as the test set.

The Human Communication Research Center Map Task Corpus (HCRC, also known as the Edinburgh Map Task) (Thompson et al., 1996) is used as the unlabeled target domain set. HCRC consists of 128 dialogs. In each dialog, both participants had a map of the same area, but the maps differed in the landmarks featured in given locations, and participants could not see their partners' maps. One map included a route, and the holder of that map was asked to verbally guide the other participant to redraw the route on his or her map. We ignore all annotations in the corpus and only use the transcribed sentences. The full corpus contains 27 084 sentences. When one-word sentences are removed (as described below), 18 738 sentences remain. Note that this corpus shares many characteristics with the CReST corpus, but there are differences in the domain: the environments, landmarks, and the task itself are different, and in HCRC, neither participant is physically present in the mapped location. Furthermore, dialectal differences exist, in that 61 of the 64 HCRC participants were from the Glasgow, Scotland area, while the 46 CReST participants were from the US.

4.2 Parsers

Both experiments use the Berkeley Parser (Petrov et al., 2006). For parser combination, we also use the Bikel Parser (Bikel, 2004) and LoPar (Schmid, 2000), and for co-training, the Stanford Parser (Klein and Manning, 2003).

Bikel's parser is a probabilistic context-free (PCFG) parser with a probability model based on Collins's model 2 (Collins, 1999); the Berkeley parser performs split-merge cycles on the training data to automatically induce a PCFG with optimized syntactic categories. Collins' model 2 (Collins, 1997) is a generative model based on bigram probabilities, dependencies between pairs of words, as well as sub-categorization frames for head-words. LoPar was used in concert with these two parsers for the parser combination experiments due to its human accessible grammar files: rule counts can be directly modified and new rules added to the LoPar grammar. Thus, we can add partially agreeing sentences, in the form of individual rules, from the HCRC data. For the co-training experiments, we used the Stanford parser instead of Bikel's because the co-training experiments require the parsers to generate confidence scores for each parse. The Berkeley parser produces such scores, Bikel's does not. The Berkeley parser's training was limited to 5 split-merge cycles in order to avoid overfitting to the PTB.

In all experiments, sentences longer than 40 words were excluded from training and testing. All parsers were trained on the source domain training sets of PTB sections 2-11 and 2-21. All experiments use gold POS tags for the PTB and CReST. HCRC is tagged with TnT (Brants, 2000), trained on the full PTB.

4.3 Parser Combination

The three parsers were used to parse the HCRC corpus. Agreement among the three was determined by bracketing alone (unlabeled condition), and bracketing along with node labels (labeled condition). For the unlabeled condition, the labels to add to training are simply taken from the parser with the highest overall baseline, i.e. Berkeley.

LoPar alone was chosen as our test parser for the experiments that involve adding agreeing rules directly to the training. For the other experiments, we also used the Berkeley parser and Bikel's parser as final parsers.

4.4 Co-Training

For co-training, the value of the n best sentences added to the training set per cycle was chosen between 20 and 500, and a minimum of four co-training cycles were performed. The size of the pool of randomly selected sentences to parse, m, was chosen from values ranging from 250 to 1500. Optimal combinations of n and m were determined by a non-exhaustive search on the PTB 2-11 training set and the CReST development set. The optimal values for n and m were found to be 20 and 500, respectively. Then, we repeated the experiment with the PTB 2-21 training set.

We used a single training set for both parsers; i.e., after each cycle, the n-best parsed sentences from each parser were added to a common training set, rather than passed to a unique training set for the opposite parser. Initial experiments showed that the set of n-best ranked sentences was comprised almost entirely of single-word sentences, leading to a decrease in performance from the baselines. Consequently, we removed all one-word sentences from the raw target domain data.

4.5 Evaluation

For evaluation, we used the standard evalb software[1] and report F_1-scores, based on labeled precision and recall. We performed significance tests using Dan Bikel's Randomized Parsing Evaluation Comparator[2].

5 Results

5.1 Parser Combination

For the experiments on parser combination, we report three baselines, one baseline per parser. Then, we investigate agreement across 3 parsers and across 2 parsers.

Agreement across 3 parsers. Here, we report results for the following experiments:

1. SENTLAB adds HCRC sentences on which the 3 parsers agree on labeled analyses.
2. SENTUNLAB adds HCRC sentences on which the 3 parsers agree on bracketing but not necessarily on labels.
3. RULES adds individual context free rules to training on which the 3 parsers agree.

The third condition can only be used with LoPar as the final parser, the other two conditions are used in combination with each parser.

In table 1, we present the results for these experiments. We also experimented with conditions where we removed one-word HCRC sentences from the additional training data. However, the F-scores with one-word sentences removed were very close to their counterparts, if not somewhat lower. For this reason, we do not report them.

[1]`http://nlp.cs.nyu.edu/evalb/`

[2]`http://www.cis.upenn.edu/~dbikel/software.html#comparator`

Experiment	sec. 2-11	sec. 2-21
Berkeley baseline	71.30	**72.24**
Bikel baseline	**71.93**	71.94
LoPar baseline	70.41	70.75
Berk.+SENTUNLAB	65.86	69.46
Berk.+SENTLAB	70.49	69.41
Bikel+SENTUNLAB	67.16	69.36
Bikel+SENTLAB	68.04	68.64
Lo.+SENTUNLAB	70.37	*72.15*
Lo.+SENTLAB	70.50	71.42
Lo.+RULES	*70.58*	71.37

Table 1: Results of the parser combination on the CReST test set (F_1). We report labeled F_1.

The results show that the baseline parsers profit only marginally from the larger training set in the second column. Note that the results are lower than normally reported for in-domain parsing. This is due to the fact that the two domains are very different. LoPar performs lower than its two counterparts, the Berkeley parser and Bikel's, as is expected, since it is a PCFG parser with a straightforward probability model.

When we add the training data from HCRC to the source training, both the Berkeley parser and Bikel's parser degrade in performance while LoPar's performance increases over its baseline. A major source of error lies in CReST's many one-word sentences: 1 638 out of 4 518. In CReST, the vast majority (1 580) of the one-word sentence parses have `INTJ` as the unary node. The extended grammars used by LoPar closely matches this distribution, with a majority of the one-word sentence parses being dominated by the `INTJ` unary node. The Berkeley and Bikel parsers, in contrast, have a strong preference to label the unary nodes as `FRAG`. Despite being trained on the same additional data, LoPar is not as subject to this errant distribution. This may be due to the fact that the probability models in the Berkeley and Bikel parsers are more finely tuned to the PTB and thus more brittle to noisy data, whereas LoPar uses a simpler model and is more robust.

For LoPar, providing additional training data from HCRC in all 3 variants improves the F-scores by a small margin over its baseline, with only one exception: In the experiment where we train LoPar on the small training set and add all sentences on which all three parsers agree, we see a small loss in the F-score. The second trend that can be observed is that LoPar trained on the large source domain data set profits more from the additional target domain data than when it is trained on the smaller source domain set.

The best performing condition given the small source domain training set is the one in which we add individual rules, RULES. Given the larger source domain training set, the best performing condition is the one using sentences with unlabeled agreement, SENTUNLAB. Thus, if the parser has a solid, large grammar from the source domain, it can use the large but noisy addition to its grammar while the smaller source domain grammar requires more high quality additions. In the setting with the small source domain grammar, RULES adds 8 966 additional rules, SENTLAB adds 3 135 rules, and SENTUNLAB adds 25 764 rules. However, note that even the best performing LoPar variant cannot outperform the results by the Berkeley baseline (or the Bikel baseline, in the setting with the smaller source domain training set).

Agreement across 2 parsers. We now turn to the experiments with enforced agreement based on a dyad of parsers. In table 2, we present results from the experiments with the relaxed condition, for each possible dyad of parsers, combined with LoPar as the final parser. We also retrained the Berkeley and the Bikel parser on the extended data sets, but the results were far below the ones for LoPar. This is interesting in itself because LoPar, as the weakest baseline parser, is capable of profiting the most from the additional target domain data. We assume that this is a consequence of LoPar's simple, but robust probability model.

The best performer for both sizes of source domain data is the combination of the Berkeley parser and Bikel's in the unlabeled sentence condition (BERKELEY/BIKELSENTUNLAB), which is also the experiment where LoPar has the most additional training, adding either 50 050 rules (sec. 2-11 experiments) or 53 123 rules (sec. 2-21 experiments) (cf. 312 614 rules in sec. 2-11 baseline, 662 266 in sec. 2-21 baseline). Also worth noting is the fact that LoPar is taking training from the agreements from the *other two* parsers. LoPar profits the most from the sentences selected by the combination of parsers that have different biases. In this way, the parser combination approach is similar to co-training. Note that when we enforce agreement between two parsers only, the addi-

Experiment	F_1 (sec. 2-11)	F_1 (sec. 2-21)
Berkeley baseline	71.30	72.24
Bikel baseline	71.93	71.94
LoPar baseline	70.41	70.75
LoPar+BERKELEY/BIKELSENTLAB	71.26	72.77†
LoPar+BERKELEY/LOPARSENTLAB	70.51	71.36
LoPar+BIKEL/LOPARSENTLAB	70.15	71.10
LoPar+BERKELEY/BIKELSENTUNLAB	**73.41**†	**73.66**†
LoPar+BERKELEY/LOPARSENTUNLAB	70.43	72.22
LoPar+BIKEL/LOPARSENTUNLAB	72.87‡	73.29†
LoPar+BERKELEY/BIKELRULES	71.52	72.22
LoPar+BERKELEY/LOPARRULES	70.62	71.20
LoPar+BIKEL/LOPARRULES	70.49	71.35

Table 2: Results for LoPar with HCRC training, based on 2 parsers, on the CReST test set. †=significance at $p < 0.001$ over the best performing baseline, ‡ at $p < 0.005$.

tional training data boosts LoPar's accuracy to improve over both the Berkeley and the Bikel baselines. We also see that in this condition, there is only a minimal difference between the small and the large source domain training set.

We also looked at the influence of quantity of (additional) training data on the results. In general, more training data leads to better results. As expected, PTB sections 2-21 perform better than sections 2-11, but as more and more data is added, the results converge, leading us to the conclusion that as more reliable target domain training data is available, the size of the initial source domain training set becomes less important. However, there must be a critical mass of additional training data before results start to improve, with more data required for a smaller source domain training set. This might suggest that the other parser combinations may not have resulted in this critical mass; in other words, that the HCRC corpus may be too small as a target domain data set given a parser combination setting.

5.2 Co-Training

In the co-training setting, the additional training set is produced by two individual parsers, the Berkeley and Stanford parser. The selection of reliable sentences is based on parser confidence values, i.e., the probabilities associated with parses. The additional sentences are added in cycles. We stopped the co-training process after 10 cycles. The results on the CReST development set for 6 cycles are given in Table 3.

The results show that both parsers reach lower

Training	PTB 2-11		PTB 2-21	
Parser	Berk.	Stan.	Berk.	Stan.
Baseline	68.24	67.83	69.18	68.39
Cycle 1	68.06	67.89	**70.04**	68.40
Cycle 2	69.68	68.10	69.49	68.40
Cycle 3	69.29	68.03	68.64	68.40
Cycle 4	**70.40**	68.25	68.68	**68.51**
Cycle 5	68.35	**68.37**	69.21	68.46
Cycle 6	70.31	68.36	69.97	68.43

Table 3: F-scores for 6 cycles (development data).

baseline results on the development set than in parser combination. It is also obvious the Berkeley parser outperforms the Stanford parser and that the larger, source domain training set has only a minimal effect on parser accuracy.

For the smaller training set, the Berkeley parser reached optimal results in the fourth co-training cycle and the Stanford parser in the fifth cycle. With n set at 20, these scores represent the addition of 160 and 200 target domain sentences to the training set, respectively. For the larger training set, the Berkeley parser reached optimal performance in the first cycle, and the Stanford parser in the fourth cycle, meaning 20 and 160 sentences were added, respectively.

We then used the grammars from the optimal cycle and PTB training set in order to parse the test set using both parsers. The results of these settings, along with the parsers' baselines on the test set are shown in Table 4. These results show that both parsers reach higher F-scores than on the development set. Moreover, the development set

Training	sec. 2-11		sec. 2-21	
Parser	Berk.	Stan.	Berk.	Stan.
Baseline	71.30	70.58	72.24	71.48
Optimized	**72.11**†	70.63†	**73.11**†	71.60†

Table 4: F-scores for co-training on the test set. †=significance at $p < 0.001$ over the baseline.

scores saw significantly higher improvements; the greatest improvement overall came from the combination of Berkeley and PTB 2-11, which rose from a baseline of 68.24 to 70.40 in the fourth cycle. The best results on the test set, for both PTB training sizes, are reached by the Berkeley parser, with an F-score of 72.11 given the small training set and an F-score of 73.11 given the larger training set. The results are surprising given that only a very small number of target domain sentences was added to the source domain training set.

5.3 Discussion

We are now in a position to compare the results of the two domain adaptation methods. A first comparison shows that both methods reach a similar performance: Given the larger PTB training set, the parser combination method reaches an F-score of 73.66 while co-training reaches 73.11. However, these results are obtained by different parsers and by training on different amounts of target domain training sentences: While the parser combination approach reaches the highest results based on using LoPar, co-training favors the Berkeley parser. And while parser combination adds 15 200 sentences from the HCRC corpus (including one-word sentences), the best co-training results are reached by adding only 20 sentences. Also, the best performing parser combination took approximately 3.5 hours while the best performing co-training experiment (which took only 1 cycle) required 2.5 hours on the same cluster.

Error analysis. In examining the results of our two approaches, unsurprisingly, we found that a large proportion of the errors are related to the considerable differences between the source and target domain. Newspaper text is more formal than spontaneous dialogs. Moreover, some phenomena that occur frequently in CReST are absent or rare in the PTB training data. For example, sentence-initial "and" is a prominent feature of CReST, but naturally, not so frequent in the PTB. There are no sentences that begin with "and" in the training set, which makes them a challenge for the parsers. Thus, in our best co-training experiment, the Berkeley parser relied heavily on the generic `X` label. However, this label is not used in this context in the gold standard. Notably, the distribution of these labels in the Stanford parses as well as in the parser combination parses is similar to that of the gold standard. However, all parse models have a tendency to assume such sentences are fragmentary and thus should be grouped under the `FRAG` label.

In general, fragmentary cases, which are abundant in CReST, are difficult for parsers to learn since they often require global information to decide that a constituent is incomplete. All parsers tend to either posit an extra element `FRAG` where there should be none, or omit it when it should be there. This can have a devastating effect on the F-scores of short sentences, which are extremely frequent in CReST.

6 Conclusion and Future Work

We performed domain adaptation for constituent parsing using two different methods. Our target domain consists of spontaneous dialogues involving collaboration between speakers. In the comparison of parser combination versus co-training, both methods outperform their respective baselines, and they reach a similar performance on the test set. We can conclude that the best parser combination adds more target domain sentences to the source domain training set while the co-training technique is faster. Potentially, LoPar could also profit from the small number of sentences chosen in the co-training experiment, but we assume that their number is too small to have an effect on the rather robust probability model.

For the future, we are planning to extend our experiments: First, we are planning to add the Stanford parser to the parser combination experiments. Then, we will use both domain adaptation methods for dependency parsing. Since both the Penn Treebank and CReST are available in dependency format, we can perform these experiments on the same data sets.

Acknowledgment

This work is based on research supported by the US Office of Naval Research (ONR) Grant #N00014-10-1-0140.

References

Daniel M Bikel. 2004. *On the parameter space of generative lexicalized statistical parsing models*. Ph.D. thesis, University of Pennsylvania.

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

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 224–231, Seattle, WA.

Wenliang Chen, Youzheng Wu, and Hitoshi Isahara. 2008. Learning reliable information for dependency parsing adaptation. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 113–120, Manchester, UK.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the Eighth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 16–23, Madrid, Spain.

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

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1051–1055, Prague, Czech Republic.

Kathleen Eberhard, Hannele Nicholson, Sandra Kübler, Susan Gunderson, and Matthias Scheutz. 2010. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta.

Jenny Rose Finkel and Christopher Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, Boulder, CO.

Sally Goldman and Yan Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 327–334, Stanford University, CA.

Daisuke Kawahara and Kiyotaka Uchimoto. 2008. Learning reliability of parses for domain adaptation of dependency parsing. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP)*, Hyderabad, India.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, Sydney, Australia.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36, Los Angeles, CA.

Antonio Valerio Miceli Barone and Giuseppe Attardi. 2012. Dependency parsing domain adaptation using transductive SVM. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 55–59, Avignon, France.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL 2007 Shared Task. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic.

Kenji Sagae. 2010. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 37–44, Uppsala, Sweden.

Beatrice Santorini. 1991. Bracketing guidelines for the Penn Treebank Project. Department of Computer and Information Science, University of Pennsylvania.

Helmut Schmid. 2000. LoPar: Design and implementation. Technical report, Universität Stuttgart.

Henry Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. 1996. The HCRC Map Task Corpus: Natural dialogue for speech recognition. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ.

Hans van Halteren, Walter Daelemans, and Jakub Zavrel. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–229.