Self-Training for Parsing Learner Text

Aoife Cahill, Binod Gyawali and James V. Bruno
Educational Testing Service,
660 Rosedale Road,
Princeton, NJ 08541,
USA
{acahill, bgyawali, jbruno}@ets.org

Abstract

We apply the well-known parsing technique of self-training to a new type of text: language-learner text. This type of text often contains grammatical and other errors which can cause problems for traditional treebank-based parsers. Evaluation on a small test set of student data shows improvement over the baseline, both by training on native or non-native text. The main contribution of this paper adds additional support for the claim that the new self-trained parser has improved over the baseline by carrying out a qualitative linguistic analysis of the kinds of differences between two parsers on non-native text. We show that for a number of linguistically interesting cases, the self-trained parser is able to provide better analyses, despite the sometimes ungrammatical nature of the text.

1 Introduction

The vast majority of treebank-based parsing research assumes that the text to be parsed is well-formed. In this paper, we are concerned with parsing text written by non-native speakers of English into phrase structure trees, as a precursor for applications in automated scoring and error detection. Non-native text often contains grammatical errors ranging in severity from minor collocational differences to extremely garbled strings that are difficult to interpret. These kinds of errors are known to cause difficulty for automated analyses (De Felice and Pulman, 2007; Lee and Knutsson, 2008).

We explore a previously documented technique for adapting a state-of-the-art parser to be able to better parse learner text: domain adaptation using self-training. Self-training is a semi-supervised learning technique that relies on some labeled data to train an initial model, and then uses large amounts of unlabeled data to iteratively improve that model. Self-training was first successfully applied in the newspaper parsing domain by McClosky et al. (2006) who used the Penn Treebank WSJ as their labeled data and unlabeled data from the North American News Text corpus. Previous attempts (Charniak, 1997; Steedman et al., 2003) had not shown encouraging results, and McClosky et al. (2006) hypothesize that the gain they saw was due to the two-phase nature of the BLLIP parser used in their experiments. In a follow-up study (McClosky et al., 2008) they find that one major factor leading to successful self-training is when the process sees known words in new combinations.

2 Related Work

Foster et al. (2011) compare edited newspaper text and unedited forum posts in a self-training parsing experiment, evaluating on a treebank of informal discussion forum entries about football. They find that both data sources perform about equally well on their small test set overall, but that the underlying grammars learned from the two sources were different. Ott and Ziai (2010) apply an out-of-the-box German dependency parser to learner text and analyze the impact on down-stream semantic interpretation. They find that core functions such as subject and object can generally be reliably detected, but that when there are key elements (e.g. main verbs) missing from the sentence that the parses are less reliable. They also found that less-severe grammatical errors such as agreement did not tend to cause problems for the parser.

An alternative approach to parsing learner text is to modify the underlying dependency scheme used in parsing to account for any grammatical errors. This can be useful because it is not always clear what the syntactic analysis of ungrammatical text should be, given some scheme designed for native text. Dickinson and Ragheb (2009) present such a modified scheme for English, designed for annotating syntactic dependencies over a modified POS tagset. Dickinson and Lee (2009) retrain a Korean dependency parser, but rather than adding additional unlabeled data as we do, they modify the original annotated training data. The modifications are specifically targeted to be able to detect errors relating to Korean postpositional particles. They show that the modified parser can be useful in detecting those kinds of particle errors and in their conclusion suggest self-training as an alternative approach to parsing of learner text. A similar alternative approach is to directly integrate error detection into the parsing process (Menzel and Schröder, 1999; Vandeventer Faltin, 2003).

3 Self-training a new parser

We first describe the data that we use for both training and evaluating our parsers, and then we describe our experiments and results.

We take the standard portion of the Penn Treebank (sections 02–21) as our seed labeled data. We then compare two different unlabeled training data sets. The first data set consists of 480,000 sentences of newspaper text extracted from the LA Times portion of the North American News Corpus (NANC). The second is a corpus of non-native written English text randomly sampled from a large dataset of student essays. It consists of 480,900 sentences from 33,637 essays written as part of a test of English proficiency, usually administered to non-native college-level students. The essays have been written to 422 different prompts (topics) and so cover a wide range of vocabulary and usage. Each essay has been assigned a proficiency level (high, medium, low) by a trained human grader. 17.5% of the sentences were from low proficiency essays, 42% from medium proficiency and 40.5% from high proficiency essays.

In order to determine the optimal number of self-training iterations and carry out our final evaluations we use a small corpus of manually treebanked sentences. The corpus consists of 1,731 sentences written by secondary level students which we randomly split into a development set (865 sentences) and a test set (866 sentences). The native language of the students is unknown, but it is likely that many spoke English as their first language. In addition, this corpus had originally been developed for another purpose and therefore contains modifications that are not ideal for our experiments. The main changes are that spelling and punctuation errors were corrected before the trees were annotated (and we do not have access to the original text). Although the treebanked corpus does not align perfectly with our requirements, we believe that it is a more useful evaluation data set than any other existing treebanked corpus.

We used the Charniak and Johnson (2005) (BLLIP) parser[1] to perform the self training experiments. Our experiment is setup as follows: first we train a baseline model on the Penn Treebank WSJ data (sections 02-21). Then, iteratively, sentences are selected from the unlabeled data sets, parsed by the parser, and combined with the previously annotated data to retrain the parser. The parser also requires development data, for which we use section 22 of the WSJ data. After each iteration we evaluate the parser using our 865-sentence development set. Parser evaluation was done using the EVALB[2] tool and we report the performance in terms of F1 score.

There are two main parameters in our self-training setup: the size of the unlabeled data set added at each iteration and the weight given to the original labeled data.[3] In preliminary experiments, we found that a block size of 40,000 sentences per each iteration and a weight of 5 on the original labeled data performed best. Given our training data, and a block size of 40K, this results in 12 iterations. In each iteration, the training data consists of the PTB data repeated 5 times, plus the parsed output of previous blocks of unlabeled data.

[1] `https://github.com/BLLIP/bllip-parser`

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

[3] Note that this approach differs to that outlined in McClosky et al. (2006) who only perform one self-training iteration. It is more similar to the approach described in Reichart and Rappoport (2007).

The results of our experiments are as shown in Figure 1. Iteration 0 corresponds to the baseline parser while iterations 1–12 are the self trained parsers. We see that the F1 score of the baseline parser is 80.9%.[4] The self trained parsers have higher accuracies compared to the baseline parser starting at the first iteration. The highest score training on non-native text (82.3%) was achieved on the 11th iteration, and the highest score training on newspaper text (81.8%) was achieved on the 8th iteration. Both of these results are statistically significantly better than the baseline parser only trained on WSJ text.[5] The graph also shows that the non-native training results in slightly higher overall f-scores than the parser trained on the native data after iteration 5, however these differences are not statistically significant.

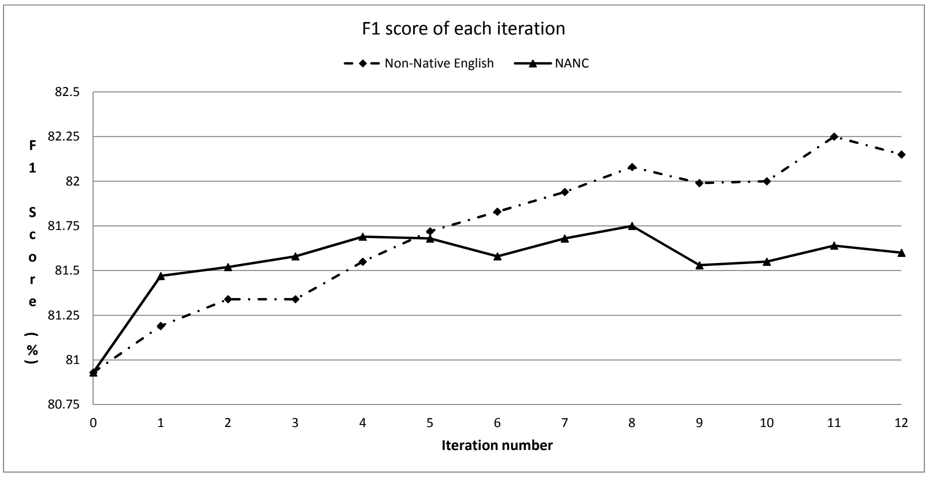


Figure 1: Performance of parsers after each iteration. Parsers used WSJ Section 22 as development data and were evaluated on the student response development data.

The final evaluation was carried out by evaluating on the student test corpus of 866 sentences, using the parsing model that performed best on the student dev corpus. The parser trained on native text achieved an f-score of 82.4% and the parser trained on the non-native text achieved an f-score of 82.6%. This difference is not statistically significant and is a similar finding to Foster et al. (2011). In another experiment, we found that if the development data used during self-training is similar to the test data, we see even smaller differences between the two different kinds of training data.[6]

4 Analysis

We carry out a qualitative analysis of the differences in parses between the original parser and one of the best-performing self-trained ones, trained on non-native text. We randomly sample 5 essays written by non-native speakers (but not overlapping with the data used to self-train the parser). Table 1 shows the number of sentences and the number of parse trees that differ, according to each proficiency level.

Proficiency	# Essays	# Sentences	# Words	# Differing Parses	% Differing Parses
High	2	30	694	12	40
Mid	1	22	389	12	54
Low	2	17	374	8	47
Totals	**5**	**69**	**1457**	**32**	**46**

Table 1: Descriptive Statistics for Essays in the Qualitative Sample

[4] Note that these overall f-scores are considerably lower than current state-of-the-art for newspaper text, indicating that this set of student texts are considerably different.

[5] Significance testing was carried out using Dan Bikel's Randomized Parsing Evaluation Comparator script for comparing `evalb` output files. We performed 1000 random shuffles and tested for p-values < 0.01.

[6] These data sets were all quite small, however, so further investigation is required to fully assess this finding.

Figure 2 reports the number of differences by proficiency level. It is important to note that these differences only included ones that were considered to be independent (e.g. a change in POS tag that necessitated a change in constituent label was only counted once). We note a trend in which the self-trained parser produces better parses than the baseline more often; however, at the highest proficiency level the baseline parser produces better parses more often. In some applications it might be possible to take the proficiency level into account before running the parser. However for many applications this will present a challenge since the parser output plays a role in predicting the proficiency level. A possible alternative would be to approximate proficiency using frequencies of spelling and other grammatical errors that can be automatically detected without relying on parser output and use this information to decide which version of the parser to use.

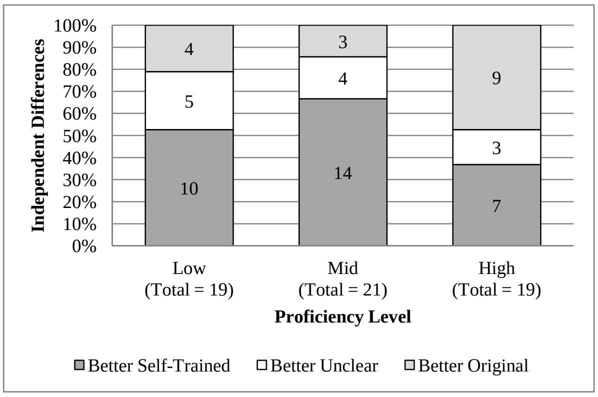


Figure 2: Unrelated Differences by Proficiency Level.

We systematically examine each of the 32 pairs of differing parse trees in the sample and manually categorize the differences. Figure 3 shows the 5 most frequent types of differences, their breakdown by proficiency level, as well as the results of a subjective evaluation on which parse was better. These judgements were made by one of the authors of this paper who is a trained linguist.

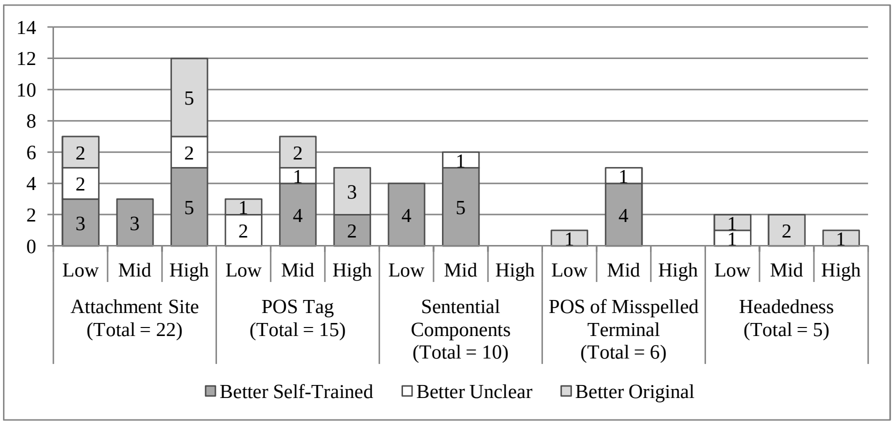


Figure 3: Parse Tree Differences by Proficiency Level.

The differences in Figure 3 are defined as follows. *Attachment Site*: the same constituent is attached to different nodes in each parse; *POS Tag*: the same terminal bears a different POS tag in each parse, where the terminal exists in our dictionary of known English[7] words; *Sentential Components*: One parse groups a set of constituents exhaustively into an S-node, while the other does not; *POS of misspelled terminal*: the same terminal bears a different POS tag in each parse, where the terminal has been flagged as a misspelling; *Headedness*: a terminal heads a maximal projection of a different syntactic category in one parse but not the other, (e.g. a VP headed by a nominal).

[7] We use the python package `enchant` with US spelling dictionaries to carry out spelling error detection.

We characterized the differences according to whether the better output was produced by the original parser, the self trained parser, or if it was not clear that either parse was better than the other. *Attachment Site* differences were evaluated according to whether or not they were attached to the constituent they modified; *POS Tag* differences were evaluated according to the Penn Treebank Guidelines (Santorini, 1995); *Sentential Components* differences were evaluated according to whether or not the terminals should indeed form a clausal constituent, infinitive, or gerund; *POS of Misspelled Terminal* differences were evaluated according to the evaluator's perception of the writer's intended target. We note that the most abundant differences are in *Attachment Site*, that the biggest improvements resulting from self-training are in the recognition of *Sentential Components* and in the identification of the *POS of Misspelled Terminals*, and that the biggest degradation is in *Headedness*.

4.1 General Difference Patterns

Using the categories defined during the manual analysis of the 5 essays, we develop rules to automatically detect these kinds of differences in a large dataset. We expect that the automatic rules will identify more differences than the linguist, however we hope to see the same general patterns. We apply our rules to an additional set of data consisting of roughly 10,000 sentences written by non-native speakers of English. Table 2 shows the number of sentences for which the parsers found different parses at each proficiency level, and Table 3 gives the totals for each of the five difference categories described above.

Proficiency	**# Essays**	**# Sentences**	**# Words**	**# Differing Parses**	**% Differing Parses**
High	256	4178	266543	2214	53
Mid	285	4168	263685	2364	57
Low	149	1657	93466	971	59
Totals	**690**	**10003**	**623694**	**5549**	**55**

Table 2: Descriptive Statistics for Essays in the Larger Sample

Difference	**Total**	**Low**	**Medium**	**High**
Attachment Site	7805	1331	3474	3000
POS Tag	6827	1205	3238	2384
Sentential Components	4103	778	1786	1539
POS of Misspelled Terminal	2040	346	894	800
Headedness	1357	353	568	436

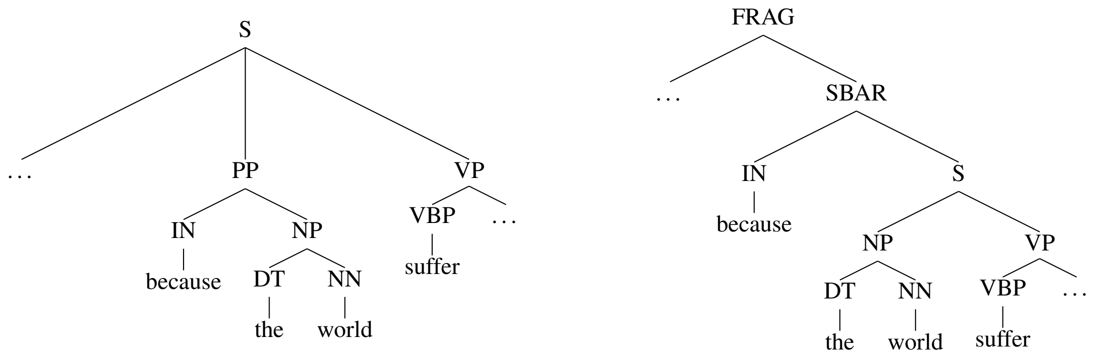
Table 3: Total number of differences detected automatically by proficiency level

We see that the proportion of sentences with different parses is similar to the 5-essay sample and also that the relative ordering of the five difference categories is identical. This at least indicates that the 5-essay sample does not differ largely in its general properties from a larger set.

4.2 Illustrative Observations

We highlight some of the most interesting differences between the baseline parser and the self-trained parser, using examples from our 5-essay sample described above.

Ambiguity of subordinating conjunctions: Figure 4 shows an example from a lower proficiency essay that contains multiple interacting differences, primarily stemming from the fact that the POS tag for a subordinating conjunction is the same as the POS tag for a regular preposition according to the Penn Treebank guidelines (Santorini, 1995). The original parser (4a) treats it as a preposition: it is dominated by PP and takes NP as a complement. The self-trained parser (4b) correctly treats *because* as a subordinating conjunction: it is dominated by SBAR and takes S as a complement. In addition, the original parser identified *suffer* as the main verb in the sentence. The self-trained parser correctly analyzes this as part of the dependent clause, however this results in no main verb being identified and an overall FRAGMENT analysis. Since it is unclear what the original intention of the writer was, this fragment analysis could be more useful for identifying grammatical errors and giving feedback.

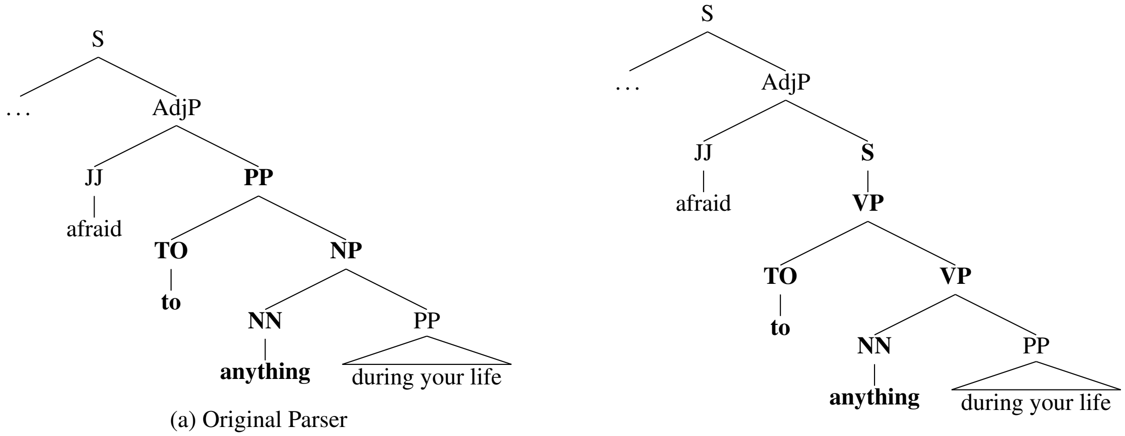


(a) Original Parser

(b) Self-Trained Parser

Figure 4: Parses for *Especaily, in this time, because the world suffer, the economy empress.*

Ambiguity of *to*: Figure 5 exemplifies a difference related to the analysis of infinitives. Here we can see that the original parser analyzed the *to* phrase as a PP (c.f. *afraid of*) whereas the self-trained parser analyzes it as an infinitival. We believe that the infinitival interpretation is slightly more likely (with a missing verb *do*), though of course it is difficult to say for sure what the intended meaning is. Here there are two interacting difference types: *Sentential Components* and *Headedness*. In the self-trained parse, *anything* is an NN that heads a VP, whereas it is an NN that appropriately heads an NP in the original parse. However, it is important to note that the self-trained parse treats *to anything* as an infinitive: a TO dominated by a VP, which is dominated by a unary-branching S. The original parse treats *to anything* as a regular PP. The fact that the self-trained parse contains a set of terminals exhaustively dominated by an S-node that does not exist in the original parse constitutes a *Sentential Components* difference. We believe that it is more useful to correctly identify infinitives and gerunds as sentential constituents, even at the cost of an XP that is apparently headed by an inappropriate terminal (VP headed by NN).



(a) Original Parser

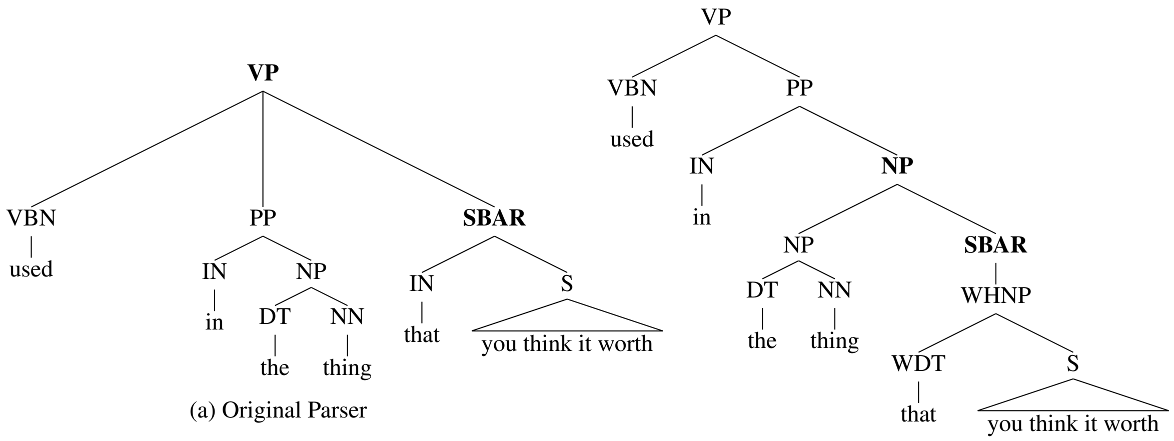
(b) Self-Trained Parser

Figure 5: Parses for *If you have this experience, you will do not afraid to anything during your life.*

Attachment ambiguity: We turn now to Figure 6. The main difference has to do with the attachment of the phrase *that you think it worth*: the SBAR is attached to the VP in the original parse (as a clausal complement) and to the NP in the self-trained parse (as a relative clause). This example also shows that a change in POS-tag can have a significant impact on the final parse tree.

5 Future Work and Conclusions

We have shown that it is possible to apply self-training techniques in order to adapt a state-of-the-art parser to be able to better parse English language learner text. We experimented with training the parser on native text as well as non-native text. In an evaluation on student data (not necessarily language-



(a) Original Parser

(b) Self-Trained Parser

Figure 6: Parses for *So I support that the money should be* *used in the thing that you think it worth.*

learner data) we found that both training sets performed at about the same level, but that both significantly out-performed the baseline parser trained only on WSJ text.

We carry out an in-depth study on a small data set of 5 learner essays and define a set of difference categories in order to describe the parse-tree differences from a linguistic perspective. We implement rules to automatically detect these parse-tree differences and show that the general proportions of errors found in the small data set are similar to that of a larger data set. We highlight some of the most interesting improvements of the parser, and we show that despite various grammatical errors present in sentences, the self-trained parser is, in general, able to assign better analyses than the baseline parser.

Of course, the self-trained parser does sometimes choose a parse that is less appropriate than the baseline one. In particular, we noticed that this happened most frequently for the highest proficiency essays. Further investigation is required to be able to better understand the reasons for this. In future work, the most informative evaluation of the self-trained parser would be in a task-based setting. We plan to investigate whether the self-trained parser improves the overall performance of tasks such as automated essay scoring or automated error detection, which internally rely on parser output.

References

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, Menlo Park, CA. AAAI Press/MIT Press.

Rachele De Felice and Stephen Pulman. 2007. Automatically Acquiring Models of Preposition Use. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 45–50, Prague, Czech Republic, June. Association for Computational Linguistics.

Markus Dickinson and Chong Min Lee. 2009. Modifying corpus annotation to support the analysis of learner language. *CALICO Journal*, 26(3):545–561.

Markus Dickinson and Marwa Ragheb. 2009. Dependency Annotation for Learner Corpora. In *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories (TLT-8)*, pages 59–70, Milan, Italy.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, and Josef van Genabith. 2011. Comparing the Use of Edited and Unedited Text in Parser Self-Training. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 215–219, Dublin, Ireland. Association for Computational Linguistics.

John Lee and Ola Knutsson. 2008. The Role of PP Attachment in Preposition Generation. In *Proceedings of CICLing 2008, 9th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 643–654, Haifa, Israel.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2008. When is Self-Training Effective for Parsing? In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 561–568, Manchester, UK, August. Coling 2008 Organizing Committee.

Wolfgang Menzel and Ingo Schröder. 1999. Error diagnosis for language learning systems. *ReCALL*, 11:20–30.

Niels Ott and Ramon Ziai. 2010. Evaluating dependency parsing performance on German learner language. In *Proceedings of the Ninth Workshop on Treebanks and Linguistic Theories (TLT-9)*, pages 175–186.

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, June. Association for Computational Linguistics.

Beatrice Santorini. 1995. Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision). Technical Report, Department of Computer and Information Science, University of Pennsylvania.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of EACL 03*, pages 331–228.

Anne Vandeventer Faltin. 2003. *Syntactic Error Diagnosis in the context of Computer Assisted Language Learning*. Ph.D. thesis, Université de Genève.