# Domain Adaptation for Dependency Parsing via Self-training

**Juntao Yu**[1], **Mohab Elkaref**[1], **Bernd Bohnet**[2]
[1] University of Birmingham, Birmingham, UK
[2]Google, London, UK
[1]{j.yu.1, m.e.a.r.elkaref}@cs.bham.ac.uk, [2]bohnetbd@google.com

## Abstract

This paper presents a successful approach for domain adaptation of a dependency parser via self-training. We improve parsing accuracy for out-of-domain texts with a self-training approach that uses confidence-based methods to select additional training samples. We compare two confidence-based methods: The first method uses the parse score of the employed parser to measure the confidence into a parse tree. The second method calculates the score differences between the best tree and alternative trees. With these methods, we were able to improve the labeled accuracy score by 1.6 percentage points on texts from a chemical domain and by 0.6 on average on texts of three web domains. Our improvements on the chemical texts of 1.5% UAS is substantially higher than improvements reported in previous work of 0.5% UAS. For the three web domains, no positive results for self-training have been reported before.

## 1 Introduction

Semi-supervised techniques gain popularity since they allow the exploitation of unlabeled data and avoid the high costs for labeling new data, cf. (Sarkar, 2001; Steedman et al., 2003; McClosky et al., 2006a; Koo et al., 2008; Søgaard and Rishøj, 2010; Petrov and McDonald, 2012; Chen et al., 2013). For domain adaptation, semi-supervised techniques have been applied successfully, cf. (Reichart and Rappoport, 2007; Petrov and McDonald, 2012; Pekar et al., 2014). Self-training is one of these appealing techniques which improves parsing accuracy by using a parser's own annotations. In a self-training iteration, a base model is first trained on annotated corpora, the base model is then used to annotate unlabeled data, finally a self-trained model is trained on both manually and automatically annotated data. This procedure might be repeated several times.

Self-training has been successfully used for instance in constituency parsing for in-domain and out-of-domain parsing (McClosky et al., 2006a; McClosky et al., 2006b; Reichart and Rappoport, 2007; Sagae, 2010). McClosky et al. (2006a) used self-training for constituency parsing. In their approaches, self-training was most effective when the parser is retrained on the combination of the initial training set and the large unlabeled dataset generated by both the generative parser and the reranker. This leads to many subsequent applications on domain adaptation via self-training for constituency parsing (McClosky et al., 2006b; Reichart and Rappoport, 2007; Sagae, 2010; Petrov and McDonald, 2012), while for dependency parsing, self-training was only effective in few cases. The question why it does not work equally well for dependency parsing is still a question that has not been satisfactorily answered. The paper tries to shed some light on the question under which circumstances and why self-training is applicable. More precisely, this paper makes the following contributions:

1. We present an effective confidence-based self-training approach.

2. We compare two confidence-based methods to select training sentences for self-training.

3. We apply our approaches on three web domains as well as on a chemical domain and we successfully improved the parsing performances for all tested domains.

The remainder of this paper is organized as follows: In Section 2, we give an overview of related work. In Section 3, we introduce two approaches



*Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10,
Bilbao, Spain; July 22–24, 2015. 

to self-training and apply those on parsing of out-of-domain data. In Section 4, we describe the data and the experimental set-up. In Section 5, we present and discuss the results. Section 6 presents our conclusions.

## 2 Related Work

Charniak (1997) applied self-training to PCFG parsing, but this first attempt to self-training for parsing failed.

Steedman et al. (2002) implemented self-training and evaluated it using several different settings. They parsed 30 sentences per iteration while the training data contained 10K sentences. Experiments with multiple iterations showed moderate improvements only which is caused probably by the small number of additional sentences used for self-training.

McClosky et al. (2006a) reported strong results with an improvement of 1.1 $F$-score using the Charniak-parser, cf. (Charniak and Johnson, 2005). McClosky et al. (2006b) applied the method later on out-of-domain texts which show good accuracy gains too.

Reichart and Rappoport (2007) showed that self-training can improve the performance of a constituency parser without a reranker when a small training set is used.

Sagae (2010) investigated the contribution of the reranker for a constituency parser. The results suggest that constituency parsers without a reranker can achieve significant improvements, but the results are still higher when a reranker is used.

In the SANCL 2012 shared task self-training was used by most of the constituency-based systems, cf. (Petrov and McDonald, 2012), which includes the top ranked system, this indicates that self-training is already an established technique to improve the accuracy of constituency parsing on out-of-domain data, cf. (Le Roux et al., 2012). However, none of the dependency-based systems used self-training in the SANCL 2012 shared task.

One of the few successful approaches to self-training for dependency parsing was introduced by Chen et al. (2008). Chen et al. (2008) improved the unlabeled attachment score about one percentage point for Chinese. Chen et al. (2008) added parsed sentences that have a high ratio of dependency edges that span only a short distance, i.e., the head and dependent are close together. The rationale for this procedure is the observation that short dependency edges show a higher accuracy than longer edges.

Kawahara and Uchimoto (2008) used a separately trained binary classifier to select reliable sentences as additional training data. Their approach improved the unlabeled accuracy of texts from a chemical domain by about 0.5%.

Goutam and Ambati (2011) applied a multi-iteration self-training approach on Hindi to improve parsing accuracy within the training domain. In each iteration, they add 1,000 additional sentences to a small initial training set of 2,972 sentences, the additional sentences were selected due to their parse scores. They improved upon the baseline by up to 0.7% and 0.4% for labeled and unlabeled attachment scores after 23 self-training iterations.

Plank (2011) applied self-training with single and multiple iterations for parsing of Dutch using the Alpino parser (Malouf and Noord, 2004), which was modified to produce dependency trees. She found self-training produces only a slight improvement in some cases but worsened when more unlabeled data was added.

Plank and Søgaard (2013) used self-training in conjunction with dependency triplets statistics and the similarity-based sentence selection for Italian out-of-domain parsing. They found that the effect of self-training is unstable and does not lead to an improvement.

Cerisara (2014) and Björkelund et al. (2014) applied self-training to dependency parsing on nine languages. Cerisara (2014) could only report negative results when they apply the self-training approach for dependency parsing. Similarly, Björkelund et al. (2014) could observe only on Swedish a positive effect.

For our approaches, confidence-based methods have been shown to be crucial such as by Dredze et al. (2008) and Crammer et al. (2009). These methods provide estimations on the quality of the predictions.

Mejer and Crammer (2012) used confidence-based methods to measure the prediction quality of a dependency parser. The confidence scores generated by these methods are correlated with the prediction accuracy of the dependency parser, i.e. higher confidence is correlated with high accuracy scores.

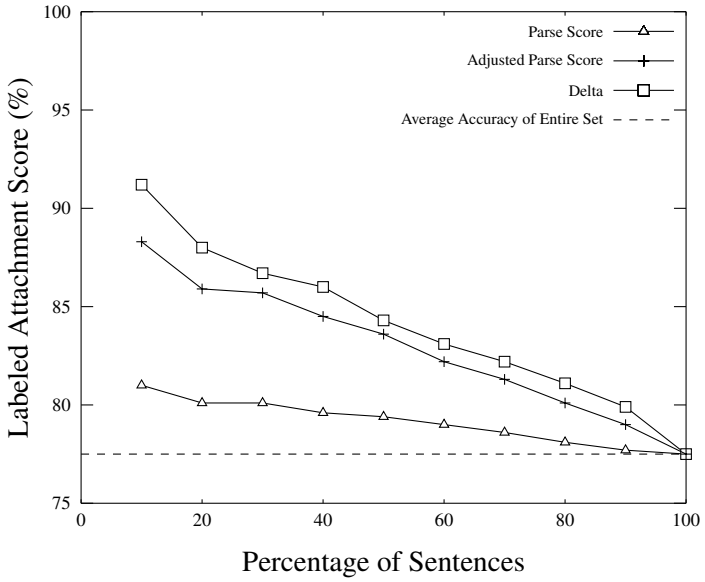


Figure 1: The graph shows the outcome of an experiment on the development set when the sentences were sorted due to the confidence score and the accuracy of the top $n$ percent is computed. The y-axis shows the accuracy and the x-axis the percentage of the number of sentences that were considered. Each curve represents a selection method.

## 3 Self-training

The hypotheses for our experiments is that the selection of high-quality dependency trees is a crucial precondition for the successful application of self-training to dependency parsing. Therefore, we explore two confidence-based methods to select such dependency trees from newly parsed sentences. Our self-training approach consists of the following steps:

1. A parser is trained on the source domain training set in order to generate a base model.

2. We analyze a large number of unlabeled sentences from a target domain with the base model.

3. We build a new training set consisting of the source domain corpus and parsed sentences that have a high confidence score.

4. We retrain the parser on the new training set in order to produce a self-trained model.

5. Finally, we use the self-trained model to parse the target domain test set.

We test two methods to gain confidence scores for a dependency tree. The first method uses the parse scores, which is based on the observation that a higher parse score is correlated with a higher parsing quality. The second method uses the method of Mejer and Crammer (2012) to compute the *Delta* score. Mejer and Crammer (2012) compute a confidence score for each edge. The algorithm attaches each edge to an alternative head. The Delta is the score difference between the original dependency tree and the tree with the changed edge. This method provides a per-edge confidence score. Note that the scores are real numbers and might be greater than 1. We changed the Delta-approach in two aspects from that of Mejer and Crammer (2012). The new parse tree contains a node that has either a different head or might have a different edge label or both since we use labeled dependency trees in contrast to Mejer and Crammer (2012). To obtain a single score for a tree, we use the averaged score of all score differences gained for each edge by the 'Delta'-method.

We use the Mate tools[1] to implement our self-training approach. The Mate tools contain a part-of-speech (pos) tagger, morphological tagger, lemmatizer, graph-based parser and an arc-standard transition-based parser. The arc-standard transition-based parser has the option to use a graph-based model to rescore the beam. The

[1] https://code.google.com/p/mate-tools/

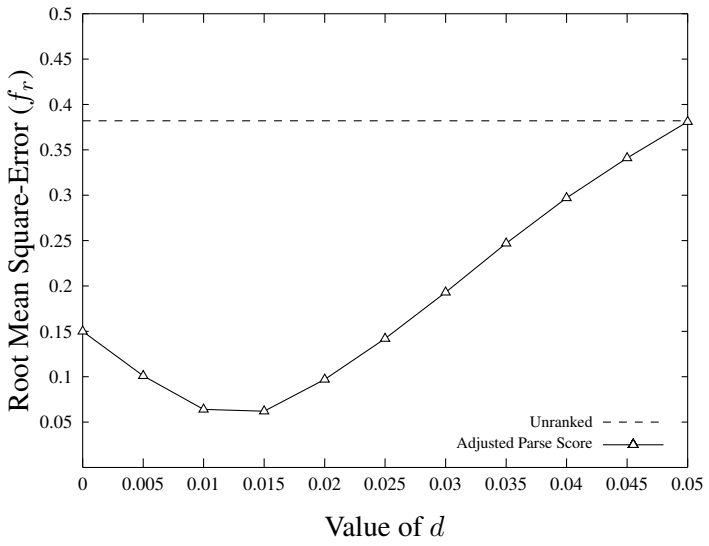


Figure 2: The root mean square-error ($f_r$) of development set after ranked by adjusted parse scores with different values of $d$.

parser has further the option to use a joint tagger and parser. The joint system is able to gain a higher accuracy for both part-of-speech tagging and parsing compared to a pipeline system.

We use the arc-standard transition-based parser which employs beam search and a graph-based rescoring model. This parser computes a score for each dependency tree by summing up the scores for each transition and dividing the score by the total number of transitions, due to the swap-operation (used for non-projective parsing), the number of transitions can vary, cf. (Kahane et al., 1998; Nivre, 2007).

Our second confidence-based method requires the computation of the score differences between the best tree and alternative trees. To compute the smallest difference (Delta), we modified the parser to derive the parse trees that contains the highest scoring alternative that replaces a given edge with an alternative one. This means either that the dependent is attached to another node or the edge label is changed, or both the dependent is attached to another node and the edge is relabeled. More precisely, during the parsing for alternative trees, beam candidates that contain the specified labeled edge will be removed from the beam at the end of each transition. Let $Score_{best}$ be the score of the best tree, $Score_i$ be the score of the alternative tree for the $i_{th}$ labeled edge and $L$ be the length of the sentence, the Delta ($Score_{Delta}$) for a parse tree is then calculated as follows:

$$Score_{Delta} = \frac{\sum_{i=1}^{L} |Score_{best} - Score_i|}{L} \quad (1)$$

To obtain high-accuracy dependency trees is crucial for our self-training approaches, thus we first assess the performance of the confidence-based methods on the development set for the selection of high-quality dependency trees. We rank the parsed sentences by their confidence scores in a descending order. Figure 1 shows the accuracy scores when selecting 10-100% of sentences with an increment of 10%. The Delta method shows the best performance for detecting high-quality parse trees, we observed that when inspecting 10% of sentences, the accuracy score difference between the Delta method and the average score of the entire set is nearly 14%. The method using the parse score does not show such a high accuracy difference. The accuracy of the 10% top ranked sentences are lower.

We observed that despite that the parse score is the averaged value of a sequence of transitions of a parse, long sentences generally exhibit a higher score. Thus, short sentences tend to be ranked in the bottom, even if they might have a higher accuracy than longer sentences. To reduce the dependency of the score on the sentence length and to maximize the correlation of the score and the accuracy, we adjust the scores for each parse tree by

| | train | test | | | dev |
|---|---|---|---|---|---|
| | **CoNLL09** | **Weblogs** | **Newsgroups** | **Reviews** | **Weblogs** |
| Sentences | 39,279 | 2,141 | 1,195 | 1,906 | 2,150 |
| Tokens | 958,167 | 40,733 | 20,651 | 28,086 | 42,144 |

Table 1: The size of the source domain training set and target domain test datasets for web domain evaluation.

| | **Weblogs** | **Newsgroups** | **Reviews** |
|---|---|---|---|
| Sentences | 513,687 | 512,000 | 512,000 |
| Tokens | 9,882,352 | 9,373,212 | 7,622,891 |

Table 2: The size of unlabeled datasets for web domain evaluation.

subtracting from them a constant $d$ multiplied by the sentence length ($L$). The new parse scores are calculated as follow:

$$Score_{adjusted} = Score_{original} - L \times d \quad (2)$$

To obtain the constant $d$, we apply the defined equation to all sentences of the development set and rank the sentences due to their adjusted scores in a descending order. The value of $d$ is selected to minimize the root mean square-error ($f_r$) of the ranked sentences. Similar to Mejer and Crammer (2012) we compute the $f_r$ by:

$$f_r = \sqrt{\sum_i n_i(c_i - a_i)^2 / (\sum_i n_i)} \quad (3)$$

We use 100 bins to divide the accuracy into ranges of one percent, parse scores in the range of $[\frac{(i-1)\times 3}{100}, \frac{i\times 3}{100}]$ are assigned to the $i_{th}$ bin[2]. Let $n_i$ be the number of sentences in $ith$ bin, $c_i$ is defined as estimated accuracy of the bin calculated by $\frac{i-0.5}{100}$ and $a_i$ is the actual accuracy of the bin. We calculate $f_r$ by iterating stepwise over $d$ from 0 to 0.05 with an increment of 0.005. Figure 2 shows the $f_r$ for the adjusted parse scores with different values of $d$. The lowest $f_r$ is achieved when $d = 0.015$, this reduce the $f_r$ from 0.15 to 0.06 when compare to the parse score method without adjustment ($d = 0$). In contrast to the $f_r = 0.06$ calculated when $d$ is set to 0.015, the unranked sentences have a $f_r$ of 0.38, which is six times larger than that of the adjusted one. The reduction on $f_r$ achieved by our adjustment indicates that the adjusted parse scores have a higher correlation to the accuracy when compare to the ones without the adjustment.

[2]We observed that parse scores computed by the parser are positive numbers and generally in the range of [0,3].

| | **train** | **test** | **unlabeled** |
|---|---|---|---|
| Sentences | 18,577 | 195 | 256,000 |
| Tokens | 446,573 | 5,001 | 6,848,072 |

Table 3: The size of datasets for chemical domain evaluation.

Figure 1 shows the performance of the adjusted parse scores for finding high accuracy parse trees in relation to the original parse score and the Delta-based method. The adjusted parse score-based method performs significantly better than that of the original score with a performance similar to the Delta method. The method based on the parse scores is faster as we do not need to apply the parser to find alternatives for each edge of a dependency tree.

## 4 Experimental Set-up

We apply the approaches on three web domains and chemical texts. Section 4.1 describes the datasets that we use in our experiments. Section 4.2 explains the parser and Section 4.3 the evaluation methods.

### 4.1 Datasets

**Web Domain.** Our experiments are evaluated on three web domains provided by Ontonotes v5.0[3] and the SANCL 2012 datasets. We use these datasets since sufficient unlabeled datasets that are required for self-training are provided by the SANCL 2012 shared task. We use the last 20%

[3]https://catalog.ldc.upenn.edu/LDC2013T19

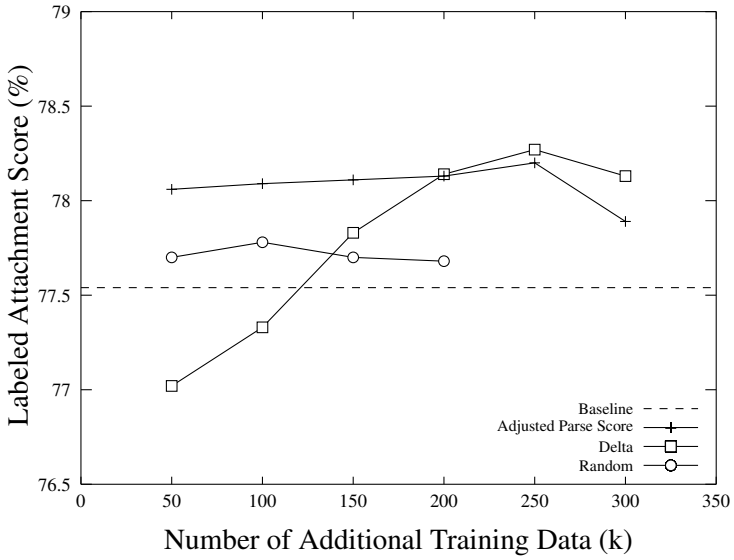


Figure 3: The effect of our self-training approaches on the weblogs development set.

of the weblogs section of the OntoNotes v5.0 corpus. Exact 50% of the selected sentences is used with SANCL Newsgroups and Reviews test data as test sets while the second half is used as a development set. We converted the datasets with the LTH constituent-to-dependency conversion tool, cf. (Johansson and Nugues, 2007). For the source domain training data, we use the CoNLL 2009 training dataset, cf. (Hajič et al., 2009). Table 1 shows the details for the training, development and test set. We use 500k of the SANCL unlabeled data for each domain after we pre-processed them by removing sentences that are longer than 500 tokens or containing non-English words which reduced the size of the datasets by 2%. Table 2 shows details about the amount of unlabeled texts.

**Chemical Domain.** To compare with previous work, we apply the approach on texts from the chemical domain that were prepared for the domain adaptation track of the CoNLL 2007 shared task, cf. (Nivre et al., 2007). Table 3 shows the details about the amount of available sentences for training, development and test set. The source data sets of the chemical domain are smaller than the ones for web domains. The training set has about half of the size. Thus we use only 250k unlabeled sentences from the chemical domain which share the same ratio of training set size to unlabeled data set size compared to the web domain data sets. To keep the same scale for training and unlabeled sets allows us easily adapt the best setting from web domain experiments.

### 4.2 Dependency Parser

We use the Mate transition-based dependency parser with default settings in our experiments, cf. Bohnet et al. (2013). For tagging, we use predicted pos tags to carry out the experiments as we believe that this is a more realistic scenario. The parser's internal tagger is used to supply the pos tags for both unlabeled sets and test datasets. In order to compare with previous work, we evaluate the approaches additionally on gold pos tags for texts of the chemical domain as gold tags were used by previous work.

The baselines are generated by training the parser on the source domain and testing the parser on the described target domain test sets.

### 4.3 Evaluation Method

For the evaluation of the parser's accuracy, we report labeled attachment scores (LAS). We included all punctuation marks in the evaluation.

For significance testing, we use the script provided by the CoNLL 2007 shared task which is Dan Bikel's randomized parsing evaluation comparator with the default settings of 10,000 iterations. The statistically significant results are marked due to their p-values, (*) p-value$<$0.05, (**) p-value$<$0.01.

## 5 Results and Discussion

**Random Selection-based Self-training.** As a baseline experiment, we apply self-training on

| | PPOS | | GPOS | |
|---|---|---|---|---|
| | **LAS** | **UAS** | **LAS** | **UAS** |
| **Parse Score** | 80.8* | 83.62* | 83.44** | 85.74** |
| **Delta** | 81.1* | 83.71* | 83.58** | 85.8** |
| **Baseline** | 79.68 | 82.5 | 81.96 | 84.28 |
| **Kawahara (Self-trained)** | - | - | - | 84.12 |
| **Kawahara (Baseline)** | - | - | - | 83.58 |
| **Sagae (Co-training)** | - | - | 81.06 | 83.42 |

Table 5: The results of the adjusted parse score-based and the Delta-based self-training approaches on the chemical test set compared with the best-reported self-training gain (Kawahara and Uchimoto, 2008) and the best results of CoNLL 2007 shared task, cf. Sagae and Tsujii (2007). (PPOS: results based on predicted pos tags, GPOS: results based on gold pos tags, Self-trained: results of self-training experiments, Co-trained: results of co-training experiments.)

| | **PS** | **Delta** | **Baseline** |
|---|---|---|---|
| **Weblogs** | 79.80** | 79.68** | 78.99 |
| **Newsgroups** | 75.88** | 75.87* | 75.3 |
| **Reviews** | 75.43* | 75.6** | 75.07 |
| **Average** | 77.03 | 77.05 | 76.45 |

Table 4: The effect of the adjusted parse score-based (PS) and the Delta-based self-training approaches on weblogs, newsgroups and reviews test sets.

randomly selected sentences that we add to the training set. Figure 3 shows an overview of the results. We obtain an improvement of 0.24% which is not statistically significant. This finding is in line with related work when applying non-confidence-based self-training approaches to dependency parsing, cf. (Cerisara, 2014; Björkelund et al., 2014).

**Parse Score-based Self-training.** For the parse score-based method, we add between 50k to 300k parsed sentences from the weblogs dataset that have been sorted by their parse scores in descending order. Figure 3 illustrates that the accuracy increase when more parsed sentences are included into the training set, we obtain the largest improvement of 0.66% when we add 250k sentences, after that the accuracy starts to decrease.

**Delta-based self-training.** For our Delta-based approach, we select additional training data with the Delta method. We train the parser by adding between 50k to 300k sentences from the target domain. We gain the largest improvement when we add 250k sentences to the training set, which improves the baseline by 0.73% (cf. Figure 3). We observe that the accuracy starts to decrease when we add 50k to 100k sentences. Our error analysis shows that these parse trees are mainly short sentences consisting of only three words. These sentences contribute probably no additional information that the parser can exploit.

**Evaluating on Test Sets.** We adapt our best settings of 250k additional sentences for both approaches and apply them to the web test sets (weblogs, newsgroups and reviews). As illustrated in Table 4, all results produced by both approaches are statistically significant improvements compared to the baseline. Our approach achieves the largest improvement of 0.81% with the parse score-based method on weblogs. For the Delta-based method, we gain the largest improvement of 0.69% on weblogs. Both approaches achieve similar improvements on newsgroups (0.57% and 0.58% for Delta and parse score-based methods, respectively). The Delta method performs better on reviews with an improvement of 0.53% vs. 0.36%. Both approaches improve on average by 0.6% on the three web domains.

We further evaluate our best settings on chemical texts provided by the CoNLL 2007 shared task. We adapt the best settings of the web domains and apply both confidence-based approaches to the chemical domain. For the constant $d$, we use 0.015 and we use 125k additional training data out of the 250k from the unlabeled data of the chemical domain. We evaluate our confidence-based methods on both predicted and gold pos tags. After retraining, both confidence-based methods achieve significant improvements in all experiments. Table 5 shows the results for the texts of the chemical domain. When we use predicted pos tags, the Delta-based method gains an improvement of

1.42% while the parse score-based approach gains 1.12%. For the experiments based on gold tags, we achieve a larger improvements of 1.62% for the Delta-based and 1.48% for the parse score-based methods.

Table 5 compares our results with that of Kawahara and Uchimoto (2008). We added also the results of Sagae and Tsujii (2007) but those are not directly comparable since they were gained with co-training. Sagae and Tsujii (2007) gained additional training data by parsing the unlabeled data with two parsers and then they select those sentence where the parsers agree.

Kawahara and Uchimoto (2008) reported positive results for self-training. They use a separate trained binary classifier to select additional training data. Kawahara and Uchimoto (2008) did evaluations only on gold pos tags. Our baseline is higher than Kawahara and Uchimoto (2008)'s self-training result, starting from this strong baseline, we could improve by 1.62% LAS and 1.52% UAS which is an error reduction of 9.6% on the UAS (cf. Table 5). The largest improvement of 1.52% compared to that of Kawahara and Uchimoto (2008) (0.54% UAS) is substantially larger. We obtained the result by a simple method and we do not need a separately trained classifier.

The confidence scores have shown to be crucial for the successful application of self-training for dependency parsing. In contrast to constituency parsing, self-training for dependency parsing does not work or at least not well without this additional confidence-based selection step. The question about a possible reason for the different behavior of self-training in dependency parsing and in constituency parsing remains open and only speculative answers could be given. We plan to investigate this further in the future.

## 6 Conclusions

In this paper, we introduced two novel confidence-based self-training approaches to domain adaptation for dependency parsing. We compared a self-training approach that uses random selection and two confidence-based approaches. While the random selection-based self-training method did *not* improve the accuracy which is in line with previously published negative results, the two confidence-based methods were able to gain statistically significant improvements and show a relative high accuracy gain.

The two confidence-based approaches achieve statistically significant improvements on all four test domains which are weblogs, newsgroups, reviews and the chemical domain. In the web domains, we gain up to 0.8 percentage points and on average both approaches improve the accuracy by 0.6%. In the chemical domain, the Delta-based and the parse score-based approaches gain 1.42% and 1.12% respectively when using predicted pos tags. When we use gold pos tags, both approaches achieved a larger improvement of 1.62% with the Delta method and 1.48% with the parse score method. In total, our approaches achieve significantly better accuracy for all four domains.

We conclude from the experiments that self-training based on confidence is worth applying in a domain adaptation scenario and that a confidence-based self-training approach seems to be crucial for the successful application of self-training in dependency parsing. This paper underlines the finding that the preselection of parse trees is probably a precondition that self-training becomes effective in the case of dependency parsing and to reach a significant accuracy gain.

## Acknowledgments

We would like to thank John Barnden for discussions and comments as well as the anonymous reviewers for their helpful reviews.

## References

Anders Björkelund, Özlem Çetinoğlu, Agnieszka Faleńska, Richárd Farkas, Thomas Mueller, Wolfgang Seeker, and Zsolt Szántó. 2014. The IMS-Wrocław-Szeged-CIS entry at the SPMRL 2014 Shared Task: Reranking and Morphosyntax meet Unlabeled Data. In *Proc. of the Shared Task on Statistical Parsing of Morphologically Rich Languages*.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richard Farkas, Filip Ginter, and Jan Hajia. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.

Christophe Cerisara. 2014. Semi-supervised experiments at LORIA for the SPMRL 2014 Shared Task. In *Proc. of the Shared Task on Statistical Parsing of Morphologically Rich Languages*, Dublin, Ireland, August.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*,

ACL '05, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. *AAAI/IAAI*, 2005:598–603.

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-Volume 1*, pages 113–120. Association for Computational Linguistics.

Wenliang Chen, Min Zhang, and Yue Zhang. 2013. Semi-supervised feature transformation for dependency parsing. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1303–1313. Association for Computational Linguistics.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems*, pages 414–422.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM.

Rahul Goutam and Bharat Ram Ambati. 2011. Exploring self training for hindi dependency parsing. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, volume 2, pages 22–69.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 1–18.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *16th Nordic Conference of Computational Linguistics*, pages 105–112. University of Tartu.

Sylvain Kahane, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity: A polynomially parsable non-projective dependency grammar. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and the 17th International Conference on Computational Linguistics (COLING)*, pages 646–652.

Daisuke Kawahara and Kiyotaka Uchimoto. 2008. Learning reliability of parses for domain adaptation of dependency parsing. In *IJCNLP*, volume 8.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 595–603.

Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. Dcu-paris13 systems for the sancl 2012 shared task.

Robert Malouf and Gertjan Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *In Proc. of IJCNLP-04 Workshop Beyond Shallow Analyses*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics.

Avihai Mejer and Koby Crammer. 2012. Are you sure?: Confidence in prediction of dependency tree edges. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 573–576, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 Shared Task of EMNLP-CoNLL 2007*, pages 915–932.

Joakim Nivre. 2007. Incremental non-projective dependency parsing. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, pages 396–403.

Viktor Pekar, Juntao Yu, Mohab El-karef, and Bernd Bohnet. 2014. Exploring options for fast domain adaptation of dependency parsers. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 54–65, Dublin, Ireland, August. Dublin City University.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Barbara Plank and Anders Søgaard. 2013. Experiments in newswire-to-law adaptation of graph-based dependency parsers. In *Evaluation of Natural Language and Speech Tools for Italian*, pages 70–76. Springer Berlin Heidelberg.

Barbara Plank. 2011. *Domain Adaptation for Parsing*. Ph.D. thesis, University of Groningen.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL*, volume 7, pages 616–623.

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 of EMNLP-CoNLL 2007*, pages 1044–1050.

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. Association for Computational Linguistics.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 175–182.

Anders Søgaard and Christian Rishøj. 2010. Semi-supervised dependency parsing using generalized tri-training. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1065–1073, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Steedman, Steven Baker, Jeremiah Crim, Stephen Clark, Julia Hockenmaier, Rebecca Hwa, Miles Osborne, Paul Ruhlen, and Anoop Sarkar. 2002. Semi-supervised training for statistical parsing.

Mark Steedman, Rebecca Hwa, Miles Osborne, and Anoop Sarkar. 2003. Corrected co-training for statistical parsers. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 95–102.