Dependency Parser Adaptation with Subtrees from Auto-Parsed Target Domain Data

Xuezhe Ma
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
xzma@uw.edu

Fei Xia
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
fxia@uw.edu

Abstract

In this paper, we propose a simple and effective approach to domain adaptation for dependency parsing. This is a feature augmentation approach in which the new features are constructed based on subtree information extracted from the auto-parsed target domain data. To demonstrate the effectiveness of the proposed approach, we evaluate it on three pairs of source-target data, compared with several common baseline systems and previous approaches. Our approach achieves significant improvement on all the three pairs of data sets.

1 Introduction

In recent years, several dependency parsing algorithms (Nivre and Scholz, 2004; McDonald et al., 2005a; McDonald et al., 2005b; McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010; Ma and Zhao, 2012) have been proposed and achieved high parsing accuracies on several treebanks of different languages. However, the performance of such parsers declines when training and test data come from different domains. Furthermore, the manually annotated treebanks that these parsers rely on are highly expensive to create. Therefore, developing dependency parsing algorithms that can be easily ported from one domain to another—say, from a resource-rich domain to a resource-poor domain—is of great importance.

Several approaches have been proposed for the task of parser adaptation. McClosky et at. (2006) successfully applied self-training to domain adaptation for constituency parsing using the reranking parser of Charniak and Johnson (2005). Reichart and Rappoport (2007) explored self-training when the amount of the annotated data is small and achieved significant improvement. Zhang and Wang (2009) enhanced the performance of dependency parser adaptation by utilizing a large-scale hand-crafted HPSG grammar. Plank and van Noord (2011) proposed a data selection method based on effective measures of domain similarity for dependency parsing.

There are roughly two varieties of domain adaptation problem—fully supervised case in which there are a small amount of labeled data in the target domain, and semi-supervised case in which there are no labeled data in the target domain. In this paper, we present a parsing adaptation approach focused on the fully supervised case. It is a feature augmentation approach in which the new features are constructed based on subtree information extracted from the auto-parsed target domain data. For evaluation, we run experiments on three pairs of source-target domains—WSJ-Brown, Brown-WSJ, and WSJ-Genia. Our approach achieves significant improvement on all these data sets.

2 Our Approach for Parsing Adaptation

Our approach is inspired by Chen et al. (2009)'s work on semi-supervised parsing with additional subtree-based features extracted from unlabeled data and by the feature augmentation method proposed by Daume III (2007). In this section, we first summarize Chen et al.'s work and explain how we extend that for domain adaptation. We will then highlight the similarity and difference between our work and Daume's method.

2.1 Semi-supervised parsing with subtree-based features

One of the most well-known semi-supervised parsing methods is self-training, where a parser trained from the labeled data set is used to parse unlabeled data, and some of those auto-parsed data are added to the labeled data set to retrain the pars-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 585–590,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

ing models. Chen et al. (2009)'s approach differs from self-training in that partial information (i.e., subtrees), instead of the entire trees, from the auto-parsed data is used to re-train the parsing models.

A subtree is a small part of a dependency tree. For example, a first-order subtree is a single edge consisting of a head and a dependent, and a second-order sibling subtree is one that consists of a head and two dependents. In Chen et al. (2009), they first extract all the subtrees in auto-parsed data and store them in a list L_{st}. Then they count the frequency of these subtrees and divide them into three groups according to their levels of frequency. Finally, they construct new features for the subtrees based on which groups they belongs to and retrain a new parser with feature-augmented training data.[1]

2.2 Parser adaptation with subtree-based Features

Chen et al. (2009)'s work is for semi-supervised learning, where the labeled training data and the test data come from the same domain; the subtree-based features collected from auto-parsed data are added to all the labeled training data to retrain the parsing model. In the supervised setting for domain adaptation, there is a large amount of labeled data in the source domain and a small amount of labeled data in the target domain. One intuitive way of applying Chen's method to this setting is to simply take the union of the labeled training data from both domains and add subtree-based features to all the data in the union when re-training the parsing model. However, it turns out that adding subtree-based features to only the labeled training data in the target domain works better. The steps of our approach are as follows:

1. Train a baseline parser with the small amount of labeled data in the target domain and use the parser to parse the large amount of unlabeled sentences in the target domain.

2. Extract subtrees from the auto-parsed data and add subtree-based features to the labeled training data in the target domain.

3. Retrain the parser with the union of the labeled training data in the two domains, where the instances from the target domain are augmented with the subtree-based features.

To state our feature augmentation approach more formally, we use X to denote the input space, and D^s and D^t to denote the labeled data in the source and target domains, respectively. Let $X^{'}$ be the augmented input space, and Φ^s and Φ^t be the mappings from X to $X^{'}$ for the instances in the source and target domains respectively. The mappings are defined by Eq 1, where $\mathbf{0} =< 0, 0, \ldots, 0 >\in X$ is the zero vector.

$$\begin{aligned} \Phi^s(\boldsymbol{x}_{org}) &= < \boldsymbol{x}_{org}, \mathbf{0} > \\ \Phi^t(\boldsymbol{x}_{org}) &= < \boldsymbol{x}_{org}, \boldsymbol{x}_{new} > \end{aligned} \tag{1}$$

Here, $\boldsymbol{x}_{org}$ is the original feature vector in X, and $\boldsymbol{x}_{new}$ is the vector of the subtree-based features extracted from auto-parsed data of the target domain. The subtree extraction method used in our approach is the same as in (Chen et al., 2009) except that we use different thresholds when dividing subtrees into three frequency groups: the threshold for the high-frequency level is TOP 1% of the subtrees, the one for the middle-frequency level is TOP 10%, and the rest of subtrees belong to the low-frequency level. These thresholds are chosen empirically on some development data set.

The idea of distinguishing the source and target data is similar to the method in (Daume III, 2007), which did feature augmentation by defining the following mappings:[2]

$$\begin{aligned} \Phi^s(\boldsymbol{x}_{org}) &= < \boldsymbol{x}_{org}, \mathbf{0} > \\ \Phi^t(\boldsymbol{x}_{org}) &= < \boldsymbol{x}_{org}, \boldsymbol{x}_{org} > \end{aligned} \tag{2}$$

Daume III showed that differentiating features from the source and target domains improved performance for multiple NLP tasks. The difference between that study and our approach is that our new features are based on subtree information instead of copies of original features. Since the new features are based on the subtree information extracted from the auto-parsed target data, they represent certain properties of the target domain and that explains why adding them to the target data works better than adding them to both the source and target data.

3 Experiments

For evaluation, we tested our approach on three pairs of source-target data and compared it with

[1] If a subtree does not appear in L_{st}, it falls to the fourth group for "unseen subtrees".

[2] The mapping in Eq 2 looks different from the one proposed in (Daume III, 2007), but it can be proved that the two are equivalent.

several common baseline systems and previous approaches. In this section, we first describe the data sets and parsing models used in each of the three experiments in section 3.1. Then we provide a brief introduction to the systems we have reimplemented for comparison in section 3.2. The experimental results are reported in section 3.3.

3.1 Data and Tools

In the first two experiments, we used the Wall Street Journal (WSJ) and Brown (B) portions of the English Penn TreeBank (Marcus et al., 1993). In the first experiment denoted by “WSJ-to-B”, WSJ corpus is used as the source domain and Brown corpus as the target domain. In the second experiment, we use the reverse order of the two corpora and denote it by “B-to-WSJ”. The phrase structures in the treebank are converted into dependencies using Penn2Malt tool[3] with the standard head rules (Yamada and Matsumoto, 2003).

For the WSJ corpus, we used the standard data split: sections 2-21 for training and section 23 for test. In the experiment of B-to-WSJ, we randomly selected about 2000 sentences from the training portion of WSJ as the labeled data in the target domain. The rest of training data in WSJ is regarded as the unlabeled data of the target domain.

For Brown corpus, we followed Reichart and Rappoport (2007) for data split. The training and test sections consist of sentences from all of the genres that form the corpus. The training portion consists of 90% (9 of each 10 consecutive sentences) of the data, and the test portion is the remaining 10%. For the experiment of WSJ-to-B, we randomly selected about 2000 sentences from training portion of Brown and use them as labeled data and the rest as unlabeled data in the target domain.

In the third experiment denoted by ’“WSJ-to-G”, we used WSJ corpus as the source domain and Genia corpus (G)[4] as the target domain. Following Plank and van Noord (2011), we used the training data in CoNLL 2008 shared task (Surdeanu et al., 2008) which are also from WSJ sections 2-21 but converted into dependency structure by the LTH converter (Johansson and Nugues, 2007). The Genia corpus is converted to CoNLL format with LTH converter, too. We randomly selected about 1000 sentences from the training portion of Genia data and use them as the labeled data of the target domain, and the rest of training data of Genia as the unlabeled data of the target domain. Table 1 shows the number of sentences of each data set used in the experiments.

	Source	Target		
	training	training	unlabeled	test
WSJ-to-B	39,832	2,182	19,632	2,429
B-to-WSJ	21,814	2,097	37,735	2,416
WSJ-to-G	39,279	1,024	13,302	1,360

Table 1: The number of sentences for each data set used in our experiments

The dependency parsing models we used in this study are the graph-based first-order and second-order sibling parsing models (McDonald et al., 2005a; McDonald and Pereira, 2006). To be more specific, we use the implementation of MaxParser[5] with 10-best MIRA (Crammer et al., 2006; McDonald, 2006) learning algorithm and each parser is trained for 10 iterations. The feature sets of first-order and second-order sibling parsing models used in our experiments are the same as the ones in (Ma and Zhao, 2012). The input to MaxParser are sentences with Part-of-Speech tags; we use gold-standard POS tags in the experiments.

Parsing accuracy is measured with unlabeled attachment score (UAS) and the percentage of complete matches (CM) for the first and second experiments. For the third experiment, we also report labeled attachment score (LAS) in order to compare with the results in (Plank and van Noord, 2011).

3.2 Comparison Systems

For comparison, we re-implemented the following well-known baselines and previous approaches, and tested them on the three data sets:

SrcOnly: Train a parser with the labeled data from the source domain only.

TgtOnly: Train a parser with the labeled data from the target domain only.

Src&Tgt: Train a parser with the labeled data from the source and target domains.

Self-Training: Following Reichart and Rappoport (2007), we train a parser with the union of the source and target labeled data, parse the unlabeled data in the target domain,

[3] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

[4] Genia distribution in Penn Treebank format is available at http://bllip.cs.brown.edu/download/genia1.0-division-rel1.tar.gz

[5] http://sourceforge.net/projects/maxparser/

add the entire auto-parsed trees to the manually labeled data in a single step without checking their parsing quality, and retrain the parser.

Co-Training: In the co-training system, we first train two parsers with the labeled data from the source and target domains, respectively. Then we use the parsers to parse unlabeled data in the target domain and select sentences for which the two parsers produce identical trees. Finally, we add the analyses for those sentences to the union of the source and target labeled data to retrain a new parser. This approach is similar to the one used in (Sagae and Tsujii, 2007), which achieved the highest scores in the domain adaptation track of the CoNLL 2007 shared task (Nivre et al., 2007).

Feature-Augmentation: This is the approach proposed in (Daume III, 2007).

Chen et al. (2009): The algorithm has been explained in Section 2.1. We use the union of the labeled data from the source and target domains as the labeled training data. The unlabeled data needed to construct subtree-based features come from the target domain.

Plank and van Noord (2011): This system performs data selection on a data pool consisting of large amount of labeled data to get a training set that is similar to the test domain. The results of the system come from their paper, not from the reimplementation of their system.

Per-corpus: The parser is trained with the large training set from the target domain. For example, for the experiment of WSJ-to-B, all the labeled training data from the Brown corpus is used for training, including the subset of data which are treated as unlabeled in our approach and other comparison systems. The results serve as an upper bound of domain adaptation when there is a large amount of labeled data in the target domain.

3.3 Results

Table 2 illustrates the results of our approach with the first-order parsing model in the first and second experiments, together with the results of the comparison systems described in section 3.2. The results with the second-order sibling parsing model is shown in Table 3. The superscript s, t and T indicates from which domain the labeled data are used in training: tag s refers to the labeled data in the source domain, tag t refers to the small amount of labeled data in the target domain, and tag T indicates that all the labeled training data from the target domain, including the ones that are treated as unlabeled in our approach, are used for training.

	WSJ-to-B		**B-to-WSJ**	
	UAS	**CM**	**UAS**	**CM**
SrcOnly^s	88.8	43.8	86.3	26.5
TgtOnly^t	86.6	38.8	88.2	29.3
Src&Tgt^{s,t}	89.1	44.3	89.4	31.2
Self-Training^{s,t}	89.2	45.1	89.8	32.1
Co-Training^{s,t}	89.2	45.1	89.8	32.7
Feature-Aug^{s,t}	89.1	45.1	89.8	32.8
Chen (2009)^{s,t}	89.3	45.0	89.7	31.8
this paper^{s,t}	**89.5**	**45.5**	**90.2**	**33.4**
Per-corpus^T	89.9	47.0	92.7	42.1

Table 2: Results with the first-order parsing model in the first and second experiments. The superscript indicates the source of labeled data used in training.

	WSJ-to-B		**B-to-WSJ**	
	UAS	**CM**	**UAS**	**CM**
SrcOnly^s	89.8	47.3	88.0	30.4
TgtOnly^t	87.7	42.2	89.7	34.2
Src&Tgt^{s,t}	90.2	48.2	90.9	36.6
Self-Training^{s,t}	90.3	48.8	91.0	37.1
Co-Training^{s,t}	90.3	48.5	90.9	38.0
Feature-Aug^{s,t}	90.0	48.4	91.0	37.4
Chen (2009)^{s,t}	90.3	49.1	91.0	37.6
this paper^{s,t}	**90.6**	**49.6**	**91.5**	**38.8**
Per-corpus^T	91.1	51.1	93.6	47.9

Table 3: Results with the second-order sibling parsing model in the first and second experiments.

Table 4 shows the results in the third experiment with the first-order parsing model. We also include the result from (Plank and van Noord, 2011), which use the same parsing model as ours. Note that this result is not comparable with other numbers in the table as it uses a larger set of labeled data, as indicated by the † superscript.

All three tables show that our system outperforms the comparison systems in all three

	WSJ-to-G	
	UAS	**LAS**
SrcOnly^s	83.8	82.0
TgtOnly^t	87.0	85.7
Src&Tgt^{s,t}	87.2	85.9
Self-Training^{s,t}	87.3	86.0
Co-Training^{s,t}	87.3	86.0
Feature-Aug^{s,t}	87.9	86.5
Chen (2009)^{s,t}	87.5	86.2
this paper^{s,t}	**88.4**	**87.1**
Plank (2011)†	-	86.8
Per-corpus^T	90.5	89.7

Table 4: Results with first-order parsing model in the third experiment. "Plank (2011)" refers to the approach in Plank and van Noord (2011).

experiments.[6] The improvement of our approach over the feature augmentation approach in Daume III (2007) indicates that adding subtree-based features provides better results than making several copies of the original features. Our system outperforms the system in (Chen et al., 2009), implying that adding subtree-based features to only the target labeled data is better than adding them to the labeled data in both the source and target domains.

Considering the three steps of our approach in Section 2.2, the training data used to train the parser in Step 1 can be from the target domain only or from the source and target domains. Similarly, in Step 3 the subtree-based features can be added to the labeled data from the target domain only or from the source and target domains. Therefore, there are four combinations. Our approach is the one that uses the labeled data from the target domain only in both steps, and Chen's system uses labeled data from the source and target domains in both steps. Table 5 compares the performance of the final parser in the WSJ-to-Genia experiment when the parser is created with one of the four combinations. The column label and the row label indicate the choice in Step 1 and 3, respectively. The table shows the choice in Step 1 does not have a significant impact on the performance of the final models; in contrast, the choice in Step 3 does matter— adding subtree-based features to the labeled data in the target domain only is much better than adding features to the data in both domains.

	TgtOnly	Src&Tgt
TgtOnly	88.4/87.1	88.4/87.1
Src&Tgt	87.6/86.3	87.5/86.2

Table 5: The performance (UAS/LAS) of the final parser in the WSJ-to-Genia experiment when different training data are used to create the final parser. The column label and row label indicate the choice of the labeled data used in Step 1 and 3 of the process described in Section 2.2.

4 Conclusion

In this paper, we propose a feature augmentation approach for dependency parser adaptation which constructs new features based on subtree information extracted from auto-parsed data from the target domain. We distinguish the source and target domains by adding the new features only to the data from the target domain. The experimental results on three source-target domain pairs show that our approach outperforms all the comparison systems.

For the future work, we will explore the potential benefits of adding other types of features extracted from unlabeled data in the target domain. We will also experiment with various ways of combining our current approach with other domain adaptation methods (such as self-training and co-training) to further improve system performance.

References

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CONLL*, pages 957–961.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine-grained n-best parsing and discriminative reranking. In *Proceedings of the 43rd Meeting of the Association for Computional Linguistics (ACL 2005)*, pages 132–139.

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 570–579, Singapore, August.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Jornal of Machine Learning Research*, 7:551–585.

[6] The results of Per-corpus are better than ours but it uses a much larger labeled training set in the target domain.

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

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA*, Tartu, Estonia.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of 48th Meeting of the Association for Computional Linguistics (ACL 2010)*, pages 1–11, Uppsala, Sweden, July.

Xuezhe Ma and Hai Zhao. 2012. Fourth-order dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 785–796, Mumbai, India, December.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 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 2006)*, pages 337–344, Sydney, Australia, July.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of European Association for Computational Linguistics (EACL-2006)*, pages 81–88, Trento, Italy, April.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL-2005)*, pages 91–98, Ann Arbor, Michigan, USA, June 25-30.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language (HLT/EMNLP 05)*, pages 523–530, Vancouver, Canada, October.

Ryan McDonald. 2006. *Discriminative learning spanning tree algorithm for dependency parsing*. Ph.D. thesis, University of Pennsylvania.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *Proceedings of the 20th international conference on Computational Linguistics (COLING'04)*, pages 64–70, Geneva, Switzerland, August 23-27.

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 Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech, June.

Barbara Plank and Gertjan van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 1566–1576, Portland, Oregon, USA, June.

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 (ACL-2007)*, pages 616–623, Prague, Czech Republic, June.

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, June.

Mihai Surdeanu, Richard Johansson, Adam Meyers, L-luis Marquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 159–177, Manchester, UK, Augest.

Hiroyasu Yamada and Yuji Matsumoto. 2003. S-tatistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT-2003)*, pages 195–206, Nancy, France, April.

Yi Zhang and Rui Wang. 2009. Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009)*, pages 378–386, Suntec, Singapore, August.