Experiments in newswire-to-law adaptation of graph-based dependency parsers

Barbara Plank[1]* and Anders Søgaard[2]

[1] University of Trento,
Via Sommarive 14, 38123 Povo, Italy
[2] University of Copenhagen,
Njalsgade 140-142, 2300 KBH S Copenhagen, Danmark
barbara.plank@disi.unitn.it, soegaard@hum.ku.dk

Abstract. We evaluate two very different methods for domain adaptation of graph-based dependency parsers on the EVALITA 2011 Domain Adaptation data, namely instance-weighting [10] and self-training [9, 6]. Since the source and target domains (newswire and law, respectively) were very similar, instance-weighting was unlikely to be efficient, but some of the semi-supervised approaches led to significant improvements on development data. Unfortunately, this improvement did not carry over to the released test data.

Keywords: Dependency Parsing; Domain Adaptation

1 Domain adaptation

In parsing it is usually assumed that training and test data are sampled from the same underlying distribution. This is almost never the case, but in some cases differences cannot be ignored. If the training and test data are sampled from similar resources, say newswire, supervised approaches to learning can induce knowledge from the training data that generalizes to the test data, but if resources differ more radically, e.g. in genre or topic, the training data may introduce a considerable *sample bias* leading to poor performance on test data.

Strategies to automatically correct sample bias in natural language processing include feature-based approaches [1, 4], instance weighting [3, 8, 10] and using semi-supervised learning algorithms [9, 2]. Most attempts to use feature-based approaches in parsing have failed, and in our experiments we therefore focused on instance weighting and semi-supervised learning algorithms.

Experiments were carried out on the official EVALITA 2011 Domain Adaptation data. The training data consists of 71,568 tokens of manually annotated Italian newswire (from the ISST-TANL corpus). The development data is a small amount of annotated sentences from the TRG corpus of Italian legislative text (5,165 tokens), and the unlabeled data was also sampled from this corpus.

* Most of the work presented here was carried out when the first author was still working for the University of Groningen, The Netherlands.

2 System description

2.1 Base parser

To find a competitive base parser, we evaluated MSTParser[3], MaltParser[4] and Mate-Tools[5] with different parameter settings. Optimizing for LAS **excluding punctuation**[6] on target domain development data, we selected the second-order projective MSTParser with 2-best MIRA learning as our base parser [7]. The LAS of the optimized MSTParser on the target development data was 79.6%. Mate-Tools equaled its performance, whereas MaltParser performed slightly worse with the official parameter setting for Italian (the CoNLL 2007 Shared Task). The parameter settings for MSTParser were confirmed by cross-validation experiments on source domain data. Our experiments also confirmed that there potentially was a lot to gain from combining the three parsers. In particular, an oracle that always relies on the two *graph-based* dependency parsers, MSTParser and Mate-Tools, in every attachment decision would lead to an LAS of 83.0% on target development data. Finally, we improved a bit on the MSTParser, correcting a few inconsistencies and adding an extra feature template, obtaining a baseline LAS of 79.7% on target domain development data. In particular we added a template that indicates dependency edges for sibling notes, since the annotation distinguishes coordination types (disjunctions and conjunctions).

2.2 Instance weighting

The intuition in instance weighting is to weight each data point in the labeled source data by the probability it was sampled from the target domain [12]. A data point that could just as well have been from the target domain is given more weight, while characteristic data points in the source domain are suppressed. In parsing, a data point is a sentence, and we implement instance weighting for structured prediction by weighting the MIRA loss function in our graph-based dependency parser [10]. To approximate the probability that a data point is sampled from a domain, we use a trigram-based logistic regression text classifier.[7]

2.3 Semi-supervised approaches

Using word clusters Our first experiment applies the simple semi-supervised approach in [5] to the evaluation campaign data set. Clusters were induced from the unlabeled target domain data using a hierarchical agglomerative clustering algorithm,[8] and we used full paths and shortened paths in the hierarchical clustering to present word clusters at different granularities. We tried to integrate

[3] http://sourceforge.net/projects/mstparser/
[4] http://maltparser.org
[5] http://code.google.com/p/mate-tools/
[6] The official results include punctuation, but we ignored it during development.
[7] http://mallet.cs.umass.edu/
[8] http://cs.stanford.edu/∼pliang/software/brown-cluster-1.2.zip

the word clusters in the feature model in different ways, but none of these attempts led to improvements. This is in a way surprising, since the unlabeled data consists of 13M words, which should be enough to induce relevant distributional similarities among words.

Dependency triplets statistics Instead of using word clusters we experimented with using dependency triplets (labeled head-dependent pairs) from auto-parser data for clustering. We used our base parser to parse the unlabeled data and calculated normalized point-wise mutual information scores for each triplet [11] , e.g.:

0.698647615001	mod	Parlamento Europeo
0.698611453092	mod	triennio 1999-2001
0.698608390934	prep	senza interruzione
0.6986066067	prep	dopo parola

We used lemmas to compute scores for triplets, but also report on an experiments with word forms (Table 2). The triplets are integrated by adding new features for every major POS tag and relation. For example for obj(drink, milk), a new feature z(v,noun) is added, whose score is the normalized point-wise mutual information between 'drink' and 'milk' with an object relation. See [11] for details. The nmpi scores range from 0 to 1, but we bin the floats into binary features. Note that features do not refer to lexical items, and the increase in model size is minimal.

Combining dependency triplets statistics with self-training Self-training is perhaps the simplest approach to automatically correcting sample selection bias, but it very rarely works. In self-training, a parser is trained on the available labeled data, in our case the source domain data, and applied to target data. The automatically labeled target data is then added to the labeled data on which the parser is re-trained. Our results below show that self-training does not lead to improvements and seems relatively unstable. However, self-training does help our parser enriched with information from auto-parsed dependency triplets.

Self-training with Jensen-Shannon divergence In self-training the parser augmented with dependency triplets statistics we also experimented with using Jensen-Shannon divergence to estimate the similarity of unlabeled data to the actual target data, selecting only the data that was most similar to the target distribution. Effects were unclear.

Co-training This approach is inspired by [9]. Training our base parser on all sentences the MSTParser and the MaltParser agree on (exact matches) as well as the original source data never led to improvements over our base parser, but using MSTParser and Mate-Tools led to small improvements. Using the non-optimized base parser (79.6%), we obtained an LAS of 80.2% selecting only

agreed-upon sentences of length 10 to 50. The difference in UAS was significant, whereas $p \sim 0.06$ for LAS. In fact using only agreed-upon target data almost equaled baseline performance (79.2%).

3 Results

Recall that all results are reported **excluding punctuation**. In the tables "nf" refers to MSTParser with our new feature template, and "nfd" adds direction to the template. The final baseline results are presented in Table 1. We also report model size (α).

Table 1. Results MST with training-k:2 with latest target devel data (corrections released on Sep 23). Excluding punctuation.

		LAS	UAS	LA	α
source devel	2o.proj.org	80.97	86.86	87.48	7557716
	2o.proj.nf	81.86	87.89	88.35	7558253
	2o.proj.nfd	**82.18**	88.46	88.53	7558645
target devel	2o.proj.org	79.36	83.82	88.72	7557716
	2o.proj.nf	**79.71**	84.36	89.36	7558253
	2o.proj.nfd	79.53	84.11	89.34	7558645

Since the domain difference between newswire and law is relatively small, we did not expect much from instance weighting. Interestingly, our text classifier seemed to discriminate well between the two kinds of text when trained on additional unlabeled data,[9] but as expected, the probabilities did not seem to correct the sample bias in the labeled source domain data.

The results using dependency triplets statistics are presented in Table 2. The suffix 'th' is the frequency threshold. The best results were obtained using statistics from all dependency triplets that were observed more than five times (136,707 triplets). We also tried only using dependency triplets from sentences that MSTParser and Mate-Tools agreed upon, but results degraded a bit.

Note that the best result this far on the target development data is 80.5%. We then turned to self-training with and without dependency triplets statistics. Without triplets self-training hurts considerably, but we obtained our best result (81.5%) on the target development data self-training a model with dependency triplets statistics using 12,800 unlabeled target domain sentences; see Figure 1. We see a slow decline with increasing amounts of unlabeled data. We did not try to balance labeled and unlabeled data by instance weighting [6].

The results using Jensen-Shannon divergence for selecting unlabeled data are also presented in Figure 1. We see an improvement over the base parser, but also a drop in accuracy around 800 sentences of unlabeled target domain data.

[9] We used the Oxford University Corpus of Italian Newspapers for the source domain and a sample of the unlabeled target domain data provided by the organizers.

Table 2. Results MST with training-k:2. Excluding punctuation.

target devel	LAS	UAS	LA	α
baseline 2o.proj.nfd	79.53	84.11	89.34	7558645
mst.100.unique.baseline.nfd.npmi.th5 (136707)	**80.54**	85.25	90.25	7559524
mst.100.unique.baseline.nfd.npmi.wordform.th5 (170857)	79.88	84.52	89.65	7559398
mst.100.unique.baseline.nfd.npmi.th10 (79454)	80.31	84.92	90.02	7559431
mst.100.unique.baseline.nfd.npmi.th20 (45045)	80.14	84.77	89.79	7559349

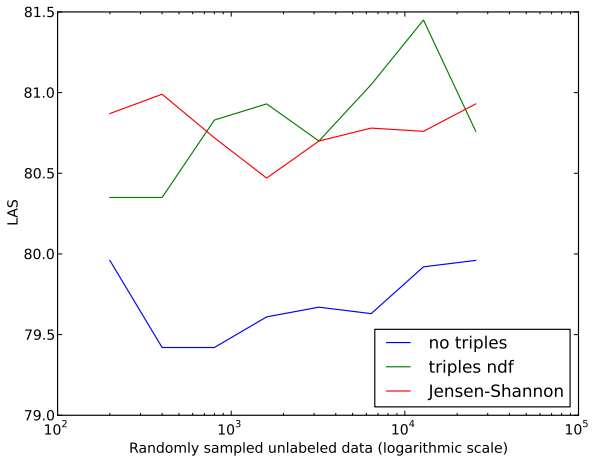

Fig. 1. Self-training with and without dependency triplets statistics.

We finally report on our co-training results. The improved MSTParser and Mate-Tools parser agreed on 58,482 unique sentences in the unlabeled target domain data. We experimented with using all sentences and only sentences of length 10 to 50 (16,436 sentences), in conjunction with the labeled source data. Co-training also led to improvements over the base parser.

Table 3. Co-training: the MSTParser trained on source data and unlabeled data agreed upon by two diverse parsers.

	LAS	UAS	LA	α
mst-mate.10-50	80.23	84.98	89.75	18875140
mst-mate.all	80.31	84.98	89.79	24432809

Finally, we experimented with combinations of the above systems, but none of our experiments led to results that were better than what could be obtained with self-training and dependency triplets statistics alone.

Test results We submitted results using dependency triplets statistics (th=5) and using self-training and dependency triplets statistics (12,800 sent.). Unfortunately, the significant improvement we observed on development data did not carry over to test data, where our final systems were slightly less accurate than our base parser.

4 Discussion

Somewhat surprisingly very few of the methods that have been previously proposed in the literature seem to be efficient on the evaluation campaign data set, including [9, 5, 8, 10]. Some of our experiments led to significant or near-significant improvements on development data, but the same set-ups led to poor results on test data. This suggests that we over-fitted our models on the small amount of development data, but it also leads us to think that there is an additional bias in the test data, not related to the marginal distribution of the unlabeled data provided by the organizers. This is supported by the following observation: The topic model Jensen-Shannon divergence [8] between the development data and the unlabeled data was 0.26, whereas the divergence between the development data and the test data was 0.35. For comparison, the divergence between development and training data was 0.47. The test data was thus half-way between the source domain and the target domain potentially leading to over-adaptation of the learned parsing model.

Acknowledgements

The research described in this paper has been partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under the grants #247758: EternalS – Trustworthy Eternal Systems via Evolving Software, Data and Knowledge, and #288024: LiMoSINe – Linguistically Motivated Semantic aggregation engiNes and by the Italian Ministry of Instruction, University and Research (MIUR), Research Program of National Interest (PRIN 2008): Parli – The Portal for Italian Natural Language Processing.

References

1. Blitzer, J., McDonald, R., Pereira, F. Domain Adaptation with Structural Correspondence Learning. In *EMNLP* (2006).
2. Chen, M., Weinberger, K., Blitzer, J. Co-training for Domain Adaptation. In *NIPS* (2011).
3. Dahlmeier, D., Ng, H.T. Domain Adaptation for Semantic Role Labeling in the Biomedical Domain. *Bioinformatics*, 26:1091–1097 (2010).

4. Daume III, H. Frustratingly Easy Domain Adaptation. In *ACL* (2007).
5. Koo, T., Carreras, X., Collins, M. Simple Semi-supervised Dependency Parsing. In *ACL* (2008).
6. McClosky, D., Charniak, E., Johnson, M. Automatic Domain Adaptation for Parsing. In *NAACL-HLT* (2010).
7. McDonald, R., Lerman, K., Pereira, F. Multilingual Dependency Analysis with a Twostage Discriminative Parser. In *CoNLL*, New York, NY (2006).
8. Plank, B., van Noord, G. Effective Measures of Domain Similarity for Parsing. In *ACL* (2011).
9. Sagae, K., Tsuji, J. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In *EMNLP-CoNLL* (2007).
10. Søgaard, A., Haulrich, M. Sentence-level Instance-weighting for Graph-based and Transition-based Dependency Parsing. In *IWPT* (2011).
11. van Noord, G. Using Self-trained Bilexical Preferences to Improve Disambiguation Accuracy. In *IWPT* (2007).
12. Zadrozny, B. Learning and Evaluating Classifiers Under Sample Selection Bias. In *ICML* (2004).