Frustratingly Hard Domain Adaptation for Dependency Parsing

Mark Dredze[1] and **John Blitzer**[1] and **Partha Pratim Talukdar**[1] and
Kuzman Ganchev[1] and **João V. Graça**[2] and **Fernando Pereira**[1]

[1]CIS Dept., University of Pennsylvania, Philadelphia, PA 19104
{mdredze|blitzer|partha|kuzman|pereira}@seas.upenn.edu

[2]L[2]F – INESC-ID Lisboa/IST, Rua Alves Redol 9, 1000-029, Lisboa, Portugal
javg@l2f.inesc-id.pt

Abstract

We describe some challenges of adaptation in the 2007 CoNLL Shared Task on Domain Adaptation. Our error analysis for this task suggests that a primary source of error is differences in annotation guidelines between treebanks. Our suspicions are supported by the observation that no team was able to improve target domain performance substantially over a state of the art baseline.

1 Introduction

Dependency parsing, an important NLP task, can be done with high levels of accuracy. However, adapting parsers to new domains *without* target domain labeled training data remains an open problem. This paper outlines our participation in the 2007 CoNLL Shared Task on Domain Adaptation (Nivre et al., 2007). The goal was to adapt a parser trained on a single source domain to a new target domain using only unlabeled data. We were given around 15K sentences of labeled text from the Wall Street Journal (WSJ) (Marcus et al., 1993; Johansson and Nugues, 2007) as well as 200K unlabeled sentences. The development data was 200 sentences of labeled biomedical oncology text (BIO, the ONCO portion of the Penn Biomedical Treebank), as well as 200K unlabeled sentences (Kulick et al., 2004). The two test domains were a collection of medline chemistry abstracts (pchem, the CYP portion of the Penn Biomedical Treebank) and the Child Language Data Exchange System corpus (CHILDES) (MacWhinney, 2000; Brown, 1973). We used the second order two stage parser and edge labeler of McDonald et al. (2006), which achieved top results in the 2006 CoNLL-X shared task. Preliminary experiments indicated that the edge labeler was fairly robust to domain adaptation, lowering accuracy by 3% in the development domain as opposed to 2% in the source, so we focused on unlabeled dependency parsing.

Our system did well, officially coming in 3rd place out of 12 teams and within 1% of the top system (Table 1). [1] In unlabeled parsing, we scored 1st and 2nd on CHILDES and pchem respectively. However, our results were obtained *without adaptation*. Given our position in the ranking, this suggests that no team was able to significantly improve performance on either test domain beyond that of a state-of-the-art parser.

After much effort in developing adaptation methods, it is critical to understand the causes of these negative results. In what follows, we provide an error analysis that attributes domain loss for this task to a difference in annotation guidelines between domains. We then overview our attempts to improve adaptation. While we were able to show limited adaptation on reduced training data or with first-order features, no modifications improved parsing with all the training data and second-order features.

2 Parsing Challenges

We begin with an error analysis for adaptation between WSJ and BIO. We divided the available WSJ data into a train and test set, trained a parser on the train set and compared errors on the test set and BIO. Accuracy dropped from 90% on WSJ to 84% on BIO. We then computed the fraction of errors involving each POS tag. For the most common

[1]While only 8 teams participated in the closed track with us, our score beat all of the teams in the open track.

Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 1051–1055,
Prague, June 2007.

	pchem l	*pchem ul*	*childes ul*	*bio ul*
Ours	80.22	83.38	61.37	83.93
Best	81.06	83.42	61.37	-
Mean	73.03	76.42	57.89	-
Rank	3rd	2nd	1st	-

Table 1: Official labeled (l) and other unlabeled (ul) submitted results for the two test domains (pchem and childes) and development data accuracy (bio). The parser was trained on the provided WSJ data.

POS types, the loss (difference in source and target error) was: verbs (2%), conjunctions (5%), digits (23%), prepositions (4%), adjectives (3%), determiners (4%) and nouns (9%). [2] Two POS types stand out: digits and nouns. Digits are less than 4% of the tokens in BIO. Errors result from the BIO annotations for long sequences of digits which do not appear in WSJ. Since these annotations are new with respect to the WSJ guidelines, it is impossible to parse these without injecting knowledge of the annotation guidelines. [3] Nouns are far more common, comprising 33% of BIO and 30% of WSJ tokens, the most popular POS tag by far. Additionally, other POS types listed above (adjectives, prepositions, determiners, conjunctions) often attach to nouns. To confirm that nouns were problematic, we modified a first-order parser (no second order features) by adding a feature indicating correct noun-noun edges, forcing the parser to predict these edges correctly. Adaptation performance rose on BIO from 78% without the feature to 87% with the feature. This indicates that most of the loss comes from missing these edges.

The primary problem for nouns is the difference between structures in each domain. The annotation guidelines for the Penn Treebank flattened noun phrases to simplify annotation (Marcus et al., 1993), so there is no complex structure to NPs. Kübler (2006) showed that it is difficult to compare the Penn Treebank to other treebanks with more complex noun structures, such as BIO. Consider the WSJ phrase “the New York State Insurance Department”. The annotation indicates a flat structure, where every token is headed by “Department”. In contrast, a similar BIO phrase has a very different structure, pursuant to the BIO guidelines. For “the detoxication enzyme glutathione transferase P1-1”, “enzyme” is the head of the NP, “P1-1” is the head of “transferase”, and “transferase” is the head of “glutathione”. Since the guidelines differ, we observe *no* corresponding structure in the WSJ. It is telling that the parser labels this BIO example by attaching every token to the final proper noun “P1-1”, exactly as the WSJ guidelines indicate. Unlabeled data cannot indicate that BIO uses a different standard.

Another problem concerns appositives. For example, the phrase “Howard Mosher, president and chief executive officer,” has “Mosher” as the head of “Howard” and of the appositive NP delimited by commas. While similar constructions occur in BIO, there are no commas to indicate this. An example is the above BIO NP, in which the phrase “glutathione transferase P1-1” is an appositive indicating which “enzyme” is meant. However, since there are no commas, the parser thinks “P1-1” is the head. However, there are not many right to left attaching nouns.

In addition to a change in the annotation guidelines for NPs, we observed an important difference in the distribution of POS tags. NN tags were almost twice as likely in the BIO domain (14% in WSJ and 25% in BIO). NNP tags, which are close to 10% of the tags in WSJ, are nonexistent in BIO (.24%). The cause for this is clear when the annotation guidelines are considered. The proper nouns in WSJ are names of companies, people and places, while in BIO they are names of genes, proteins and chemicals. However, for BIO these nouns are labeled NN instead of NNP. This decision effectively removes NNP from the BIO domain and renders all features that depend on the NNP tag ineffective. In our above BIO NP example, all nouns are labeled NN, whereas the WSJ example contains NNP tags. The largest tri-gram differences involve nouns, such as *NN-NN-NN*, *NNP-NNP-NNP*, *NN-IN-NN*, and *IN-NN-NN*. However, when we examine the coarse POS tags, which do not distinguish between nouns, these differences disappear. This indicates that while the overall distribution of POS tags is similar between the domains, the fine grained tags differ. These fine grained tags provide more information than coarse tags; experiments that removed fine grained tags

[2] We measured these drops on several other dependency parsers and found similar results.

[3] For example, the phrase “(R = 28% (10/26); K=10% (3/29); chi2 test: p=0.014).”

hurt WSJ performance but did not affect BIO.

Finally, we examined the effect of unknown words. Not surprisingly, the most significant differences in error rates concerned dependencies between words of which one or both were unknown to the parser. For two words that were seen in the training data loss was 4%, for a single unknown word loss was 15%, and 26% when both words were unknown. Both words were unknown only 5% of the time in BIO, while one of the words being unknown was more common, reflecting 27% of decisions. Upon further investigation, the majority of unknown words were nouns, which indicates that unknown word errors were caused by the problems discussed above.

Recent theoretical work on domain adaptation (Ben-David et al., 2006) attributes adaptation loss to two sources: the difference in the distribution between domains and the difference in labeling functions. Adaptation techniques focus on the former since it is impossible to determine the latter without knowledge of the labeling function. In parsing adaptation, the former corresponds to a difference between the features seen in each domain, such as new words in the target domain. The decision function corresponds to differences between annotation guidelines between two domains. Our error analysis suggests that the primary cause of loss from adaptation is from differences in the annotation guidelines themselves. Therefore, significant improvements cannot be made without specific knowledge of the target domain's annotation standards. No amount of source training data can help if no relevant structure exists in the data. Given the results for the domain adaptation track, it appears no team successfully adapted a state-of-the-art parser.

3 Adaptation Approaches

We survey the main approaches we explored for this task. While some of these approaches provided a modest performance boost to a simple parser (limited data and first-order features), no method added any performance to our best parser (all data and second-order features).

3.1 Features

A natural approach to improving parsing is to modify the feature set, both by removing features less likely to transfer and by adding features that are more likely to transfer. We began with the first approach and removed a large number of features that we believed transfered poorly, such as most features for noun-noun edges. We obtained a small improvement in BIO performance on limited data only. We then added several different types of features, specifically designed to improve noun phrase constructions, such as features based on the lexical position of nouns (common position in NPs), frequency of occurrence, and NP chunking information. For example, trained on in-domain data, nouns that occur more often tend to be heads. However, none of these features transfered between domains.

A final type of feature we added was based on the behavior of nouns, adjectives and verbs in each domain. We constructed a feature representation of words based on adjacent POS and words and clustered words using an algorithm similar to that of Saul and Pereira (1997). For example, our clustering algorithm grouped first names in one group and measurements in another. We then added the cluster membership as a lexical feature to the parser. None of the resulting features helped adaptation.

3.2 Diversity

Training diversity may be an effective source for adaptation. We began by adding information from multiple different parsers, which has been shown to improve in-domain parsing. We added features indicating when an edge was predicted by another parser and if an edge crossed a predicted edge, as well as conjunctions with edge types. This failed to improve BIO accuracy since these features were less reliable at test time. Next, we tried instance bagging (Breiman, 1996) to generate some diversity among parsers. We selected with replacement 2000 training examples from the training data and trained three parsers. Each parser then tagged the remaining 13K sentences, yielding 39K parsed sentences. We then shuffled these sentences and trained a final parser. This failed to improve performance, possibly because of conflicting annotations or because of lack of sufficient diversity. To address conflicting annotations, we added slack variables to the MIRA learning algorithm (Crammer et al., 2006) used to train the parsers, without success. We measured diversity by comparing the parses of each model. The difference in annotation agreement between the three instance bagging parsers was about half the difference between these parsers and the gold annotations. While we believe this is not enough diversity, it was not feasible to repeat our experiment with a large number of parsers.

3.3 Target Focused Learning

Another approach to adaptation is to favor training examples that are *similar to* the target. We first modified the weight given by the parser to each training sentence based on the similarity of the sentence to target domain sentences. This can be done by modifying the loss to limit updates in cases where the sentence does not reflect the target domain. We tried a number of criteria to weigh sentences without success, including sentence length and number of verbs. Next, we trained a discriminative model on the provided unlabeled data to predict the domain of each sentence based on POS n-grams in the sentence. Training sentences with a higher probability of being in the target domain received higher weights, also without success. Further experiments showed that any decrease in training data hurt parser performance. It would seem that the parser has no difficulty learning important training sentences in the presence of unimportant training examples.

A related idea focused on words, weighing highly tokens that appeared frequently in the target domain. We scaled the loss associated with a token by a factor proportional to its frequency in the target domain. We found certain scaling techniques obtained tiny improvements on the target domain that, while significant compared to competition results, are not statistically significant. We also attempted a similar approach on the feature level. A very predictive source domain feature is not useful if it does not appear in the target domain. However, limiting the feature space to target domain features had no effect. Instead, we scaled each feature's value by a factor proportional to its frequency in the target domain and trained the parser on these scaled feature values. We obtained small improvements on small amounts of training data.

4 Future Directions

Given our pessimistic analysis and the long list of failed methods, one may wonder if parser adaptation is possible at all. We believe that it is. First, there may be room for adaptation with our domains if a common annotation scheme is used. Second, we have stressed that typical adaptation, modifying a model trained on the source domain, will fail but there may be unsupervised parsing techniques that improve performance after adaptation, such as a rule based NP parser for BIO based on knowledge of the annotations. However, this approach is unsatisfying as it does not allow general purpose adaptation.

5 Acknowledgments

We thank Joel Wallenberg and Nikhil Dinesh for their informative and helpful linguistic expertise, Kevin Lerman for his edge labeler code, and Koby Crammer for helpful conversations. Dredze is supported by a NDSEG fellowship; Ganchev and Talukdar by NSF ITR EIA-0205448; and Blitzer by DARPA under Contract No. NBCHD03001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

References

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *NIPS*.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

R. Brown. 1973. *A First Language: The Early Stages*. Harvard University Press.

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

R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*.

Sandra Kübler. 2006. How do treebank annotation schemes influence parsing results? or how not to compare apples and oranges. In *RANLP*.

S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, and L. Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proc. of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

B. MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency parsing with a two-stage discriminative parser. In *Conference on Natural Language Learning (CoNLL)*.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).*

Lawrence Saul and Fernando Pereira. 1997. Aggregate and mixed-order markov models for statistical language modeling. In *EMNLP*.