The NAIST Dependency Parser for SANCL2012 Shared Task

Katsuhiko Hayashi* and **Shuhei Kondo*** and **Kevin Duh** and **Yuji Matsumoto**
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, JAPAN
katsuhiko-h,shuhei-k,kevinduh,matsu@is.naist.jp

Abstract

This paper presents the NAIST dependency parser for the SANCL2012 "Parsing the Web" Shared Task. Our base system is an in-house shift-reduce parser. In order to robustly adapt the parser to heterogeneous web data, we enhanced it with (1) dependency-based clusters and (2) consensus labels from unlabeled corpora. We found that these two enhancements gave small but promising improvements of 0.2-0.3 unlabeled attachment score (UAS).

1 Introduction

Dependency parsing technology has advanced rapidly in the past years, with accuracies in the 90% (UAS) on English newswire, c.f. (McDonald and Pereira, 2006; Koo and Collins, 2010; Hayashi et. al., 2012). However, much work remains in parsing arbitrary text, such as blogs and reviews on the Web. We participated in the SANCL 2012 "Parsing the Web" Shared Task in order to experiment with robust adaptation of parsers. The task uses the following datasets (for details, please refer to (Petrov and McDonald, 2012)):

1. Labeled Training Data: ∼30,000 parsed sentences in Newswire domain (Ontonotes WSJ)
2. Unlabeled Data: Unlabeled sentences in five web domains (∼100,000 sentences each).
3. Dev Data: Parsed sentences in web domains of Weblog and Email (∼2,000 sentences each).
4. Test Data: Sentences from other web domains (Question Answer, Review, Newsgroups)

The goal is "general adaptation" as opposed to the more constrained "source-to-target adaptation". A single parser is trained using Datasets 1 & 2, with the hope that robust results will be achieved on all web datasets. Our base system is a shift-reduce parser. For adaptation, we enhanced this with (1) dependency-based word cluster features, and (2) consensus labels generated from the unlabeled data. These are described in the following sections, followed by experimental results and discussions.

2 Shift-Reduce Parser

Our in-house dependency parser employs the arc-standard shift-reduce algorithm with beam search and dynamic programming techniques (Huang, 2010). This algorithm is originally developed for unlabeled dependency parsing, thus we extended it to be able to handle labeled dependency parsing. When the reduce action creates an arc, our parser determines a label for the arc by a discriminative model with labeled dependency features used in MSTParser[1] (McDonald and Pereira, 2006). This is more efficient than a transition-based parser which uses all labeled reduce actions (Zhang and Joakim, 2011). The unlabeled dependency features used in our parser are mostly equivalent to those of (Huang, 2010). The unlabeled and labeled discriminative models are jointly trained by using a averaged perceptron algorithm with early update (Collins, 2004).

* The first two authors contributed equally to this work.

[1] www.seas.upenn.edu/∼strctlrn/MSTParser/MSTParser.html

Our parser uses POS tags provided by the Stanford POS tagger (Toutanova et. al., 2003). The model is trained on the provided labeled training data, with the following training options: bidirectional5words, naacl2003unknowns and wordshapes(-1,1). The number of iterations was 10. We used only the Penn Tagset and did not experiment with the Universal POS-set provided with the training data (due to time constraints).

3 Extensions for Domain Adaptation

Theoretically, we can view the domain adaptation problem as a change in the distribution $p(x, y) = p(y|x)p(x)$, where $p(x)$ is the input distribution of sentences and $p(y|x)$ is the conditional distribution of parse y given sentence x (Jiang and Zhai, 2007). Either $p(x)$ or $p(y|x)$ may vary as we move across domains, and mismatch in either (with respect to the training data) leads to performance degradation. Our dependency-based cluster tackles mismatch in $p(x)$ by tying together word features, while our consensus label approach attempts to reduce $p(y|x)$ mismatch by re-training on auto-parses of web data.

3.1 Dependency-based Cluster Features

Word clusters have been shown to give good performance for dependency parsing, especially in the context of semi-supervised parsing (Koo et. al., 2008). Here we follow Koo's approach in incorporating word clusters as additional features into the parser. However, rather than cluster using word n-gram information, we cluster using dependency n-gram information. The motivation is that head/child information may provide more useful generalizations than neighboring left/right context (Sagae and Gordon, 2009; Haffari et. al., 2011). In particular, we first parse the unlabeled data with the MST Parser, then extract head-child information as bigram dependencies. This is given to the Brown clustering algorithm[2], generating 32 cluster features for the shift-reduce parser.

3.2 Consensus Labels from Unlabeled Data

Self-training and co-training can be effective methods for domain adaptation (McClosky et. al., 2008). Here we experimented with a co-training scheme where consensus parses provided by the MST parser and Berkeley parser[3] are given to our shift-reduce parser. Specifically, we parsed the unlabeled training data with the MST parser and the Berkeley parser, both trained on the labeled training data. Then we converted the outputs of the Berkeley parser into dependency trees and extracted trees on which the two parsers reached a (exact-match) consensus in terms of the unlabeled arc structure. As a result 5,200 trees were extracted, and we added them to the training data for the submitted system. For POS tags and the edge labels, we used the outputs of the Stanford POS tagger and the MST parser.

4 Preliminary Experiments and Discussion

Tables 1 and 2 show the UAS and LAS on the Dev data, using either true (goldtag) or predicted (autotag) POS tags. The **baseline** system is our shift-reduce parser trained on the labeled newswire training data (dataset 1), without any extensions for adaptation. The **+ngramcluster** is the baseline system enhanced with ngram-based clusters like (Koo et. al., 2008), while **+depcluster** is the one enhanced with dependency-based clusters. Systems using additional consensus labels are indicated by **+consensus**. Our official submission is **baseline+consensus+depclusters**, which is the best of the bunch in general and observes a slight improvement of 0.2 UAS in EMAIL (autotag) and of 0.3 UAS in WEBLOG (autotag) over **baseline**. The test results are shown in Table 3.

Due to time constraints[4], we did not attempt many important experimental variations. Although the slight improvements are promising, we think the following future work are worth investigating in detail:

- Word clustering with combined n-gram and syntax information, since it is likely a single view is ineffective in clustering rare words.
- Consensus labels derived from partial parse matches, since exact match is too strict and generates too few additional labels.
- Analysis and quantification of $p(x)$ and $p(y|x)$ mismatches in web data. Which one is more serious and deserves more research?

[2] cs.stanford.edu/∼pliang/software/brown-cluster-1.2.zip

[3] http://code.google.com/p/berkeleyparser/

[4] We had a late start in participating in this shared task.

	EMAIL		WEBLOGS	
SYSTEM	goldtag	autotag	goldtag	autotag
baseline	83.4	78.0	89.0	85.6
+ngramclusters	83.5	78.0	88.6	85.6
+depclusters	83.4	**78.2**	88.7	85.6
+consensus	83.1	78.1	88.6	85.5
+consensus+ngramclusters	**83.7**	78.0	88.8	85.5
+consensus+depclusters	83.6	**78.2**	**89.1**	**85.9**

Table 1: Unlabeled Attachment Score (UAS) on Dev data

	EMAIL		WEBLOGS	
SYSTEM	goldtag	autotag	goldtag	autotag
baseline	80.3	73.1	86.3	81.5
+ngramclusters	80.2	73.0	86.0	81.6
+depclusters	80.1	**73.2**	86.1	81.6
+consensus	79.9	**73.2**	86.1	81.5
+consensus+ngramclusters	**80.7**	72.9	86.2	81.5
+consensus+depclusters	80.4	**73.2**	**86.4**	**81.7**

Table 2: Labeled Attachment Score (LAS) on Dev data

	ANSWERS (A)	NEWSGROUP (B)	REVIEWS (C)	AVG (A-C)	WSJ (D)
LAS	73.54	79.83	75.72	76.36	87.95
UAS	79.89	84.59	81.99	82.16	90.99
POS	89.92	91.39	90.47	90.59	97.40

Table 3: Test Results (baseline+consensus+depclusters)

References

Michael Collins and Brian Roark. 2004. Incremental Parsing with the Perceptron Algorithm. *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume, pg. 111–118*

Katsuhiko Hayashi and Taro Watanabe and Masayuki Asahara and Yuji Matsumoto. 2012. Head-driven Transition-based Parsing with Top-down Prediction. *Proceedings of the Association for Computational Linguistics (ACL).*

Liang Huang and Kenji Sagae. 2010. Dynamic Programming for Linear-Time Incremental Parsing. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pg. 1077–1086*,

Terry Koo and Michael Collins. 2010. Efficient Third-order Dependency Parsers *Proceedings of the Association for Computational Linguistics (ACL).*

Terry Koo and Xavier Carreras and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing *Proceedings of the Association for Computational Linguistics (ACL).*

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. *Proc. of the Association for Computational Linguistics (ACL).*

Gholamreza Haffari and Marzieh Razavi and Anoop Sarkar. 2011 An Ensemble Model that Combines Syntactic and Semantic Clustering for Discriminative Dependency Parsing. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.*

David McClosky and Eugene Charniak and Mark Johnson. When is Self-Training Effective for Parsing? *Proceedings of the International Conference on Computational Linguistics (COLING 2008)*

Ryan McDonald and Fernando Pereira. 2006. Online Learning of Approximate Dependency Parsing Algorithms. *Proceedings of the 11th EACL.*

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).*

Kenji Sagae and Andrew Gordon. 2009 Clustering Words by Syntactic Similarity Improves Dependency Parsing of Predicate-Argument Structures. *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*

Kristina Toutanova and Dan Klein and Christopher Manning and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL 2003, pp. 252-259.*

Yue Zhang and Joakim Nivre. 2011. Transition-based Dependency Parsing with Rich Non-local Features. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pg. 188–193*