Dependency Language Models for Transition-based Dependency Parsing

Juntao Yu
University of Birmingham
Birmingham, UK
`j.yu.1@cs.bham.ac.uk`

Bernd Bohnet
Google
London, UK
`bohnetbd@google.com`

Abstract

In this paper, we present an approach to improve the accuracy of a strong transition-based dependency parser by exploiting dependency language models that are extracted from a large parsed corpus. We integrated a small number of features based on the dependency language models into the parser. To demonstrate the effectiveness of the proposed approach, we evaluate our parser on standard English and Chinese data where the base parser could achieve competitive accuracy scores. Our enhanced parser achieved state-of-the-art accuracy on Chinese data and competitive results on English data. We gained a large absolute improvement of one point (UAS) on Chinese and 0.5 points for English.

1 Introduction

In recent years, using unlabeled data to improve natural language parsing has seen a surge of interest as the data can easy and inexpensively be obtained, cf. (Sarkar, 2001; Steedman et al., 2003; McClosky et al., 2006; Koo et al., 2008; Søgaard and Rishøj, 2010; Petrov and McDonald, 2012; Chen et al., 2013; Weiss et al., 2015). This is in stark contrast to the high costs of manually labeling new data. Some of the techniques such as self-training (McClosky et al., 2006) and co-training (Sarkar, 2001) use auto-parsed data as additional training data. This enables the parser to learn from its own or other parser's annotations. Other techniques include word clustering (Koo et al., 2008) and word embedding (Bengio et al., 2003) which are generated from a large amount of unannotated data. The outputs can be used as features or inputs for parsers. Both groups

of techniques have been shown effective on syntactic parsing tasks (Zhou and Li, 2005; Reichart and Rappoport, 2007; Sagae, 2010; Søgaard and Rishøj, 2010; Yu et al., 2015; Weiss et al., 2015). However, most word clustering and the word embedding approaches do not consider the syntactic structures and most self-/co-training approaches can use only a relatively small additional training data as training parsers on a large corpus might be time-consuming or even intractable on a corpus of millions of sentences.

Dependency language models (DLM) (Shen et al., 2008) are variants of language models based on dependency structures. An N-gram DLM is able to predict the next child when given N-1 immediate previous children and their head. Chen et al. (2012) integrated first a high-order DLM into a second-order graph-based parser. The DLM allows the parser to explore high-order features but not increasing the time complexity. Following Chen et al. (2012), we adapted the DLM to transition-based dependency parsing. Our approach is different from Chen et al. (2012)'s in a number of important aspects:

1. We applied the DLM to a strong parser that on its own has a competitive performance.

2. We revised their feature templates to integrate the DLMs with a transition-based system and labeled parsing.

3. We used DLMs in joint tagging and parsing, and gained up to 0.4% on tagging accuracy.

4. Our approach could use not only single DLM but also multiple DLMs during parsing.

5. We evaluated additionally with DLMs extracted from higher quality parsed data which two parsers assigned the same annotations.

11

Overall, our approach improved upon a competitive baseline by 0.51% for English and achieved state-of-the-art accuracy for Chinese.

2 Related work

Previous studies using unlabeled text could be classified into two groups by how unlabeled data is used for training.

The first group uses unlabeled data (usually parsed data) directly in the training process as additional training data. The most common approaches in this group are self-/co-training. McClosky et al. (2006) applied first self-training to a constituency parser. This was later adapted to dependency parsing by Kawahara and Uchimoto (2008) and Yu et al. (2015). Compared to the self-training approach used by McClosky et al. (2006), both self-training approaches for dependency parsing need an additional selection step to predict high-quality parsed sentences for retraining. The basic idea behind this is similar to Sagae and Tsujii (2007)'s co-training approach. Instead of using a separately trained classifier (Kawahara and Uchimoto, 2008) or confidence-based methods (Yu et al., 2015), Sagae and Tsujii (2007) used two different parsers to obtain the additional training data. Sagae and Tsujii (2007) shows that when two parsers assign the same syntactic analysis to sentences then the parse trees have usually a higher parsing accuracy. Tri-training (Zhou and Li, 2005; Søgaard and Rishøj, 2010) is a variant of co-training which involves a third parser. The base parser is retrained on additional parse trees that the other two parsers agreed on.

The second group uses the unlabeled data indirectly. Koo et al. (2008) used word clusters built from unlabeled data to train a parser. Chen et al. (2008) used features extracted from short distance relations of a parsed corpus to improve a dependency parsing model. Suzuki et al. (2009) used features of generative models estimated from large unlabelled data to improve a second order dependency parser. Their enhanced models improved upon the second order baseline models by 0.65% and 0.15% for English and Czech respectively. Mirroshandel et al. (2012) used the relative frequencies of nine manually selected head-dependent patterns calculated from parsed French corpora to rescore the n-best parses. Their approach gained a labeled improvement of 0.8% over the baseline. Chen et al. (2013) combined

meta features based on frequencies with the basic first-/second-order features. The meta features are extracted from parsed annotations by counting the frequencies of basic feature representations in a large corpus. With the help of meta features, the parser achieved the state-of-the-art accuracy on Chinese. Kiperwasser and Goldberg (2015) added features based on the statistics learned from unlabeled data to a weak first-order parser and they achieved 0.7% improvement on the English data. Word embeddings that represent words as high dimensional vectors are mostly used in neural network parsers (Chen and Manning, 2014; Weiss et al., 2015) and play an important role in those parsers. The approach most close to ours is reported by Chen et al. (2012) who applied a high-order DLM to a second-order graph-based parser for unlabeled parsing. Their DLMs are extracted from an English corpus that contains 43 million words (Charniak, 2000) and a 311 million word corpus of Chinese (Huang et al., 2009) parsed by a parser. From a relatively weak baseline, additional DLM-based features gained 0.6% UAS for English and an impressive 2.9% for Chinese.

3 Our Approach

Dependency language models were introduced by Shen et al. (2008) to capture long distance relations in syntactic structures. An N-gram DLM predicts the next child based on N-1 immediate previous children and their head. We integrate DLMs extracted from a large parsed corpus into the Mate parser (Bohnet et al., 2013). We first extract DLMs from a corpus parsed by the base model. We then retrain the parser with additional DLM-based features.

Further, we experimented with techniques to improve the quality of the syntactic annotations which we use to build the DLMs. We parse the sentences with two different parsers and then select the annotations which both parsers agree on. The method is similar to co-training except that we do not train the parser directly on these sentences.

We build the DLMs with the method of Chen et al. (2012). For each child x_{ch}, we gain the probability distribution $P_u(x_{ch}|H)$, where H refers $N-1$ immediate previous children and their head x_h. The previous children for x_{ch} are those who share the same head with x_{ch} but closer to the head word according to the word sequence in the sentence. Let's consider the left side child x_{Lk} in the

	train	dev	test
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_0\_pos >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_0\_word >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_1\_pos >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_1\_word >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_0\_pos, s_1\_pos >$		
	$< NO_{DLM}, \phi(P_u(s_0)), \phi(P_u(s_1)), label, s_0\_word, s_1\_word >$		

Table 1: Feature templates which we use in the parser.

	train	dev	test
PTB	2-21	22	23
CTB5	001-815, 1001-1136	886-931, 1148-1151	816-885, 1137-1147

Table 2: Our data splits for English and Chinese

dependency relations $(x_{Lk}...x_{L1}, x_h, x_{R1}...x_{Rm})$ as an example, the N-1 immediate previous children for x_{Lk} are $x_{Lk-1}..x_{Lk-N+1}$. In our approach, we estimate $P_u(x_{ch}|H)$ by the relative frequency:

$$P_u(x_{ch}|H) = \frac{count(x_{ch}, H)}{\sum_{x'_{ch}} count(x'_{ch}, H)} \quad (1)$$

By their probabilities, the N-grams are sorted in a descending order. We then used the thresholds of Chen et al. (2012) to replace the probabilities with one of the three classes (PH, PM, PL) according to their position in the sorted list, i.e. the N-grams whose probability has a rank in the first 10% receives the tag PH, PM refers probabilities ranked between 10% and 30%, probabilities that ranked below 30% are replaced with PL. During parsing, we use an additional class PO for relations not presented in the DLM. In the preliminary experiments, the PH class is mainly filled by unusual relations that only appeared a few times in the parsed text. To avoid this, we configured the DLMs to only use elements which have a minimum frequency of three, i.e. $count(x_{ch}, H) \geq 3$. Table 1 shows our feature templates, where NO_{DLM} is an index which allows DLMs distinguish from each other, s_0, s_1 are the top and the second top of the stack, $\phi(P_u(s_0/s_1))$ refers the coarse label of probabilities $P_u(x_{s_0/s_1}|H)$ (one of the PH, PM, PL, PO), s_0/s_1_pos, s_0/s_1_word refer to the part-of-speech tag, word form of s_0/s_1, and $label$ is the dependency label between the s_0 and the s_1.

4 Experimental Set-up

For our experiments, we used the Penn English Treebank (PTB) (Marcus et al., 1993) and Chinese Treebank 5 (CTB5) (Xue et al., 2005). For English, we follow the standard splits and used Stanford parser [1] v3.3.0 to convert the constituency trees into Stanford style dependencies (de Marneffe et al., 2006). For Chinese, we follow the splits of Zhang and Nivre (2011), the constituency trees are converted to dependency relations by Penn2Malt[2] tool using head rules of Zhang and Clark (2008). Table 2 shows the splits of our data. We used gold segmentation for Chinese tests to make our work comparable with previous work. We used predicted part-of-speech tags for both languages in all evaluations. Tags are assigned by base parser's internal joint tagger trained on the training set. We report labeled (LAS) and unlabeled (UAS) attachment scores, punctuation marks are excluded from the evaluation.

For the English unlabeled data, we used the data of Chelba et al. (2013) which contains around 30 million sentences (800 million words) from the news domain. For Chinese, we used Xinhua portion of Chinese Gigaword [3] Version 5.0 (LDC2011T13). The Chinese unlabeled data we used consists of 20 million sentences which is roughly 450 million words after being segmented by ZPar[4] v0.7.5. The word segmentor is trained on the CTB5 training set. In most of our experiments, the DLMs are extracted from data annotated by our base parser. For the evaluation on higher quality DLMs, the unlabeled data is additionally tagged and parsed by Berkeley parser (Petrov and Klein, 2007) and is converted to dependency trees with the same tools as for gold data.

We used Mate transition-based parser with its default setting and a beam of 40 as our baseline.

5 Results and Discussion

Combining different N-gram DLMs. We first evaluated the effects of adding different number of DLMs. Let m be the DLMs we used in the experiments, e.g. m=1-3 refers all three (unigram, bigram and trigram) DLMs are used. We evaluate

---

with both single and multiple DLMs that extracted from 5 million sentences for both languages. We started from only using unigram DLM (m=1) and then increasing the m until the accuracy drops. Table 3 shows the results with different DLM settings. The unigram DLM is most effective for English, which improves above the baseline by 0.38%. For Chinese, our approach gained a large improvement of 1.16% with an m of 1-3. Thus, we use m=1 for English and m=1-3 for Chinese in the rest of our experiments.

Exploring DLMs built from corpora of different size and quality. To evaluate the influence of the size and quality of the input corpus for building the DLMs, we experiment with corpora of different size and quality.

We first evaluate with DLMs extracted from the different number of single-parsed sentences. We extracted DLMs start from a 5 million sentences corpus and increase the size of the corpus in step until all of the auto-parsed sentences are used. Table 4 shows our results on English and Chinese development sets. For English, the highest accuracy is still achieved by DLM extracted from 5 million sentences. While for Chinese, we gain the largest improvement of 1.2% with DLMs extracted from 10 million sentences.

We further evaluate the influence of DLMs extracted from higher quality data. The higher quality corpora are prepared by parsing unlabeled sentences with the Mate parser and the Berkeley parser and adding the sentences to the corpus where both parsers agree. For Chinese, only 1 million sentences that consist of 5 tokens in average had the same syntactic structures assigned by the two parsers. Unfortunately, this amount is not sufficient for the experiments as their average sentence length is in stark contrast with the training data (27.1 tokens). For English, we obtained 7 million sentences with an average sentence length of 16.9 tokens.

To get a first impression of the quality, we parsed the development set with the two parsers. When the parsers agree, the parse trees have an accuracy of 97% LAS, while the labeled scores of both parsers are around 91%. This indicates that parse trees where both parsers return the same tree have a higher accuracy. The DLM extracted from 7 million higher quality English sentences achieved a higher accuracy of 91.56% which outperform the baseline by 0.51%.

m	0	1	2	3	1-2	1-3	1-4
English	91.05	**91.43**	91.14	91.22	91.27	91.26	N/A
Chinese	78.95	79.85	79.42	79.06	79.97	**80.11**	79.73

Table 3: Effects (LAS) of different number of DLMs for English and Chinese. $m = 0$ refers the baseline.

Size	0	5	10	20	30
English	91.05	**91.43**	91.38	91.13	91.28
Chinese	78.95	80.11	**80.15**	79.72	N/A

Table 4: Effects (LAS) of DLMs extracted from different size (in million sentences) of corpus. Size = 0 refers the baseline.

Main Results on Test Sets. We applied the best settings tuned on the development sets to the test sets. The best setting for English is the unigram DLM derived from the double parsed sentences. Table 5 presents our results and top performing dependency parsers which were evaluated on the same English data set. Our approach with 40 beams surpasses our baseline by 0.46/0.51% (LAS/UAS) [5] and is only lower than the few best neural network systems. When we enlarge the beam, our enhanced models achieved similar improvements. Our semi-supervised result with 150 beams are more competitive when compared with the state-of-the-art. We cannot directly compare our results with that of Chen et al. (2012) as they evaluated on an old Yamada and Matsumoto (2003) format. In order to have an idea of the accuracy difference between our baseline and the second-order graph-based parser they used, we include our baseline on Yamada and Matsumoto (2003) conversion. As shown in table 5 our baseline is 0.62% higher than their semi-supervised result and this is 1.28% higher than their baseline. This confirms our claim that our baseline is much stronger.

For Chinese, we extracting the DLMs from 10 million sentences parsed by the Mate parser and using the unigram, bigram and the trigram DLMs together. Table 6 shows the results of our approach and a number of best Chinese parsers. Our system gained a large improvement of 0.93/0.98% [6] for labeled and unlabeled attachment scores when using a beam of 40. When larger beams are used our approach achieved even larger improvement of more than one percentage point for both labeled and un-

---

[5]Significant in Dan Bikel's test ($p < 10^{-3}$).
[6]Significant in Dan Bikel's test ($p < 10^{-5}$).

System	Beam	POS	LAS	UAS
Zhang and Nivre (2011)	32	97.44	90.95	93.00
Bohnet and Kuhn (2012)	80	97.44	91.19	93.27
Martins et al. (2013)	N/A	97.44	90.55	92.89
Zhang and McDonald (2014)	N/A	97.44	91.02	93.22
Chen and Manning (2014)†	1	N/A	89.60	91.80
Dyer et al. (2015)†	1	97.30	90.90	93.10
Weiss et al. (2015)†	8	97.44	92.05	93.99
Andor et al. (2016)†	32	97.44	92.79	94.61
Dozat and Manning (2017)†	N/A	N/A	94.08	95.74
Liu and Zhang (2017)†	N/A	N/A	**95.20**	**96.20**
Chen et al. (2012) Baseline *	8	N/A	N/A	92.10
Chen et al. (2012) DLM *	8	N/A	N/A	92.76
Our Baseline *	40	97.33	92.44	93.38
Our Baseline	40	97.36	90.95	93.08
	80	97.34	91.05	93.28
	150	97.34	91.05	93.29
Our DLM	40	97.38	91.41	93.59
	80	97.39	91.47	93.65
	150	97.42	91.56	93.74

Table 5: Comparing with top performing parsers on English. (* means results that are evaluated on Yamada and Matsumoto (2003) conversion. † means neural network-based parsers)

System	Beam	POS	LAS	UAS
Hatori et al. (2011)	64	93.94	N/A	81.33
Li et al. (2012)	N/A	94.60	79.01	81.67
Chen et al. (2013)	N/A	N/A	N/A	83.08
Chen et al. (2015)	N/A	93.61	N/A	82.94
Our Baseline	40	93.99	78.49	81.52
	80	94.02	78.48	81.58
	150	93.98	78.96	82.11
Our DLM	40	94.27	79.42	82.51
	80	94.39	79.79	82.79
	150	94.40	**80.21**	**83.28**

Table 6: Comparing with top performing parsers on Chinese.

labeled accuracy when compared to the respective baselines. Our scores with the default beam size (40) are competitive and are 0.2% higher than the best reported result (Chen et al., 2013) when increasing the beam size to 150. Moreover, we gained improvements up to 0.42% for part-of-speech tagging on Chinese tests.

6 Conclusion

In this paper, we applied dependency language models (DLM) extracted from a large parsed corpus to a strong transition-based parser. We integrated a small number of DLM-based features into the parser. We demonstrate the effectiveness of our DLM-based approach by applying our approach to English and Chinese. We achieved statistically significant improvements on labeled and unlabeled scores of both languages. Our parsing system improved by DLMs outperforms most of

the systems on English and is competitive. For Chinese, we gained a large improvement of one point and our accuracy is 0.2% higher than the best reported result. In addition to that, our approach gained an improvement of 0.4% on Chinese part-of-speech tagging.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 2442–2452.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3:1137–1155. http://dl.acm.org/citation.cfm?id=944919.944966.

Bernd Bohnet and Jonas Kuhn. 2012. The best of both worlds – a graph-based completion model for transition-based parsers. In *Proceedings of the 13th Conference of the European Chpater of the Association for Computational Linguistics (EACL)*. pages 77–87.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas Filip Ginter, and Jan Hajic. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics* 1.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*. pages 132–139.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Philipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *Computing Research Repository (CoRR)* abs/1312.3005:1–6.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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*. Association for Computational Linguistics, pages 113–120.

Wenliang Chen, Min Zhang, and Haizhou Li. 2012. Utilizing dependency language models for graph-based dependency parsing models. In *Proceedings of the 50th Annual Meeting of the Association for*

Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, pages 213–222.

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*. Association for Computational Linguistics, pages 1303–1313. http://aclweb.org/anthology/D13-1129.

Wenliang Chen, Min Zhang, and Yue Zhang. 2015. Distributed feature representations for dependency parsing. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 23(3):451–460. https://doi.org/10.1109/TASLP.2014.2365359.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*.

Timothy Dozat and Christopher Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations*. https://openreview.net/pdf?id=Hk95PK9le.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 334–343. http://www.aclweb.org/anthology/P15-1033.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint pos tagging and dependency parsing in chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pages 1216–1224. http://www.aclweb.org/anthology/I11-1136.

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1222–1231.

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

Eliyahu Kiperwasser and Yoav Goldberg. 2015. Semi-supervised dependency parsing using bilexical contextual features from auto-parsed data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for

Computational Linguistics, Lisbon, Portugal, pages 1348–1353. http://aclweb.org/anthology/D15-1158.

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.

Zhenghua Li, Min Zhang, Wanxiang Che, and Ting Liu. 2012. A separately passive-aggressive training algorithm for joint POS tagging and dependency parsing. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, Mumbai, India, pages 1681–1698. http://www.aclweb.org/anthology/C12-1103.

J. Liu and Y. Zhang. 2017. In-Order Transition-based Constituent Parsing. *ArXiv e-prints* .

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19:313–330.

A. Martins, M. Almeida, and N. A. Smith. 2013. "turning on the turbo: Fast third-order non-projective turbo parsers". In *Annual Meeting of the Association for Computational Linguistics - ACL*. volume -, pages 617 – 622.

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

Seyed Abolghasem Mirroshandel, Alexis Nasr, and Joseph Le Roux. 2012. Semi-supervised dependency parsing using lexical affinities. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '12, pages 777–785. http://dl.acm.org/citation.cfm?id=2390524.2390634.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*. pages 404–411.

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)*.

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. 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. Association for Computational Linguistics, pages 37–44.

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.

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.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. *ACL-08: HLT* page 577.

Anders Søgaard and Christian Rishøj. 2010. Semi-supervised dependency parsing using generalized tri-training. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 1065–1073. http://dl.acm.org/citation.cfm?id=1873781.1873901.

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.

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pages 551–560. http://www.aclweb.org/anthology/D/D09/D09-1058.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of ACL 2015*. pages 323–333.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phase structure annotation of a large corpus. *Journal of Natural Language Engineering* 11:207–238.

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

Juntao Yu, Mohab Elkaref, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*. Association for Computational Linguistics, Bilbao, Spain, pages 1–10. http://www.aclweb.org/anthology/W15-2201.

Hao Zhang and Ryan McDonald. 2014. Enforcing structural diversity in cube-pruned dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 656–661. http://www.aclweb.org/anthology/P/P14/P14-2107.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 562–571.

Yue Zhang and Joakim Nivre. 2011. Transition-based parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *Knowledge and Data Engineering, IEEE Transactions on* 17(11):1529–1541.