

Joint Inference for Heterogeneous Dependency Parsing

Guangyou Zhou and Jun Zhao

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, China
{gyzhou, jzhao}@nlpr.ia.ac.cn

Abstract

This paper is concerned with the problem of heterogeneous dependency parsing. In this paper, we present a novel joint inference scheme, which is able to leverage the consensus information between heterogeneous treebanks in the parsing phase. Different from stacked learning methods (Nivre and McDonald, 2008; Martins et al., 2008), which process the dependency parsing in a pipelined way (e.g., a second level uses the first level outputs), in our method, multiple dependency parsing models are coordinated to exchange consensus information. We conduct experiments on Chinese Dependency Treebank (CDT) and Penn Chinese Treebank (CTB), experimental results show that joint inference can bring significant improvements to all state-of-the-art dependency parsers.

1 Introduction

Dependency parsing is the task of building dependency links between words in a sentence, which has recently gained a wide interest in the natural language processing community and has been used for many problems ranging from machine translation (Ding and Palmer, 2004) to question answering (Zhou et al., 2011a). Over the past few years, supervised learning methods have obtained state-of-the-art performance for dependency parsing (Yamada and Matsumoto, 2003; McDonald et al., 2005; McDonald and Pereira, 2006; Hall et al., 2006; Zhou et al., 2011b; Zhou et al., 2011c). These methods usually rely heavily on the manually annotated treebanks for training the dependency models. However, annotating syntac-

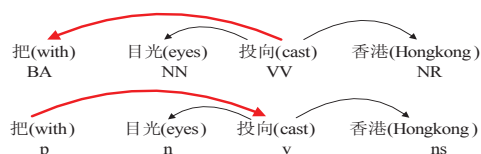


Figure 1: Different grammar formalisms of syntactic structures between CTB (upper) and CDT (below). CTB is converted into dependency grammar based on the head rules of (Zhang and Clark, 2008).

tic structure, either phrase-based or dependency-based, is both time consuming and labor intensive. Making full use of the existing manually annotated treebanks would yield substantial savings in data-annotation costs.

In this paper, we present a joint inference scheme for heterogeneous dependency parsing. This scheme is able to leverage consensus information between heterogeneous treebanks during the inference phase instead of using individual output in a pipelined way, such as stacked learning methods (Nivre and McDonald, 2008; Martins et al., 2008). The basic idea is very simple: although heterogeneous treebanks have different grammar formalisms, they share some consensus information in dependency structures for the same sentence. For example in Figure 1, the dependency structures actually share some partial agreements for the same sentence, the two words “eyes” and “Hongkong” depend on “cast” in both Chinese Dependency Treebank (CDT) (Liu et al., 2006) and Penn Chinese Treebank (CTB) (Xue et al., 2005). Therefore, we would like to train the dependency parsers on individual heterogeneous treebank and jointly parse the same sentences with consensus information exchanged between them.

The remainder of this paper is divided as fol-

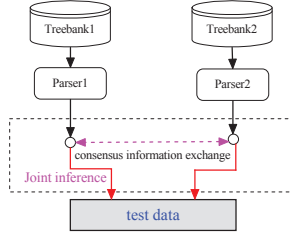


Figure 2: General joint inference scheme of heterogeneous dependency parsing.

lows. Section 2 gives a formal description of the joint inference for heterogeneous dependency parsing. In section 3, we present the experimental results. Finally, we conclude with ideas for future research.

2 Our Approach

The general joint inference scheme of heterogeneous dependency parsing is shown in Figure 2. Here, heterogeneous treebanks refer to two Chinese treebanks: CTB and CDT, therefore we have only two parsers, but the framework is generic enough to integrate more parsers. For easy explanation of the joint inference scheme, we regard a parser without consensus information as a *baseline parser*, a parser incorporates consensus information called a *joint parser*. Joint inference provides a framework that accommodates and coordinates multiple dependency parsing models. Similar to Li et al. (2009) and Zhu et al. (2010), the joint inference for heterogeneous dependency parsing consists of four components: (1) Joint Inference Model; (2) Parser Coordination; (3) Joint Inference Features; (4) Parameter Estimation.

2.1 Joint Inference Model

For a given sentence x , a joint dependency parsing model finds the best dependency parsing tree y^* among the set of possible candidate parses $\mathcal{Y}(x)$ based on a scoring function F_s :

$$y^* = \arg \max_{y \in \mathcal{Y}(x)} F_s(x, y) \quad (1)$$

Following (Li et al., 2009), we will use d_k to denote the k th joint parser, and also use the notation $\mathcal{H}_k(x)$ for a list of parse candidates of sentence x determined by d_k . The s th joint parser can be written as:

$$F_s(x, y) = P_s(x, y) + \sum_{k, k \neq s} \Psi_k(y, \mathcal{H}_k(x)) \quad (2)$$

where $P_s(x, y)$ is the score function of the s th baseline model, and each $\Psi_k(y, \mathcal{H}_k(x))$ is a partial

consensus score function with respect to d_k and is defined over y and $\mathcal{H}_k(x)$:

$$\Psi_k(y, \mathcal{H}_k(x)) = \sum_l \lambda_{k,l} f_{k,l}(y, \mathcal{H}_k(x)) \quad (3)$$

where each $f_{k,l}(y, \mathcal{H}_k(x))$ is a feature function based on a consensus measure between y and $\mathcal{H}_k(x)$, and $\lambda_{k,l}$ is the corresponding weight parameter. Feature index l ranges over all consensus-based features in equation (3).

2.2 Parser Coordination

Note that in equation (2), though the baseline score function $P_s(x, y)$ can be computed individually, the case of $\Psi_k(y, \mathcal{H}_k(x))$ is more complicated. It is not feasible to enumerate all parse candidates for dependency parsing. In this paper, we use a bootstrapping method to solve this problem. The basic idea is that we can use baseline models' n -best output as seeds, and iteratively refine joint models' n -best output with joint inference. The joint inference process is shown in Algorithm 1.

Algorithm 1 Joint inference for multiple parsers

Step1: For each joint parser d_k , perform inference with a baseline model, and memorize all dependency parsing candidates generated during inference in $\mathcal{H}_k(x)$;

Step2: For each candidate in $\mathcal{H}_k(x)$, we extract subtrees and store them in $\mathcal{H}'_k(x)$. First, we extract bigram-subtrees that contain two words. If two words have a dependency relation, we add these two words as a subtree into $\mathcal{H}'_k(x)$. Similarly, we can extract trigram-subtrees. Note that the dependency direction is kept. Besides, we also store the "ROOT" word of each candidate in $\mathcal{H}'_k(x)$;

Step3: Use joint parsers to re-parse the sentence x with the baseline features and joint inference features (see subsection 2.3). For joint parser d_k , consensus-based features of any dependency parsing candidate are computed based on current setting of $\mathcal{H}'_s(x)$ for all s but k . New dependency parsing candidates generated by d_k in re-parsing are cached in $\mathcal{H}''_k(x)$;

Step4: Update all $\mathcal{H}_k(x)$ with $\mathcal{H}''_k(x)$;

Step5: Iterate from Step2 to Step4 until a preset iteration limit is reached.

In Algorithm 1, dependency parsing candidates of different parsers can be mutually improved. For example, given two parsers d_1 and d_2 with candidates \mathcal{H}_1 and \mathcal{H}_2 , improvements on \mathcal{H}_1 enable d_2 to improve \mathcal{H}_2 , and \mathcal{H}_1 benefits from improved \mathcal{H}_2 , and so on.

We can see that a joint parser does not enlarge the search space of its baseline model, the only change is parse scoring. By running a complete inference process, joint model can be applied to re-parsing all candidates explored by a parser.

Thus Step3 can be viewed as full-scale candidates reranking because the reranking scope is beyond the limited n -best output currently cached in \mathcal{H}_k .

2.3 Joint Inference Features

In this section we introduce the consensus-based feature functions $f_{k,l}(y, \mathcal{H}_k(x))$ introduced in equation (3). The formulation can be written as:

$$f_{k,l}(y, \mathcal{H}_k(x)) = \sum_{y' \in \mathcal{H}_k(x)} P(y'|d_k) I_l(y, y') \quad (4)$$

where y is a dependency parse of x by using parser d_s ($s \neq k$), y' is a dependency parse in $\mathcal{H}_k(x)$ and $P(y'|d_k)$ is the posterior probability of dependency parse y' parsed by parser d_k given sentence x . $I_l(y, y')$ is a consensus measure defined on y and y' using different feature functions.

Dependency parsing model $P(y'|d_k)$ can be predicted by using the global linear models (GLMs) (e.g., McDonald et al. (2005); McDonald and Pereira (2006)). The consensus-based score functions $I_l(y, y')$ include the following parts:

(1) *head-modifier dependencies*. Each head-modifier dependency (denoted as “*edge*”) is a tuple $t = \langle h, m, h \rightarrow m \rangle$, so $I_{edge}(y, y') = \sum_{t \in y} \delta(t, y')$.

(2) *sibling dependencies*: Each sibling dependency (denoted as “*sib*”) is a tuple $t = \langle i, h, m, h \leftarrow i \rightarrow m \rangle$, so $I_{sib}(y, y') = \sum_{t \in y} \delta(t, y')$.

(3) *grandparent dependencies*: Each grandparent dependency (denoted as “*gp*”) is a tuple $t = \langle h, i, m, h \rightarrow i \rightarrow m \rangle$, so $I_{gp}(y, y') = \sum_{\langle h, i, m, h \rightarrow i \rightarrow m \rangle \in y} \delta(t, y')$.

(4) *root feature*: This feature (denoted as “*root*”) indicates whether the multiple dependency parsing trees share the same “*ROOT*”, so $I_{root}(y, y') = \sum_{\langle ROOT \rangle \in y} \delta(\langle ROOT \rangle, y')$.

$\delta(\cdot, \cdot)$ is an indicator function— $\delta(t, y')$ is 1 if $t \in y'$ and 0 otherwise, feature index $l \in \{edge, sib, gp, root\}$ in equation (4). Note that $\langle h, m, h \rightarrow m \rangle$ and $\langle m, h, m \rightarrow h \rangle$ are two different edges.

In our joint model, we extend the baseline features of (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007) by conjoining with the consensus-based features, so that we can learn in which kind of contexts the different parsers agree/disagree. For the third-order features (e.g., grand-siblings and tri-siblings) described in (Koo et al., 2010), we will discuss it in future work.

2.4 Parameter Estimation

The parameters are tuned to maximize the dependency parsing performance on the development set, using an algorithm similar to the average perceptron algorithm due to its strong performance and fast training (Koo et al., 2008). Due to limited space, we do not present the details. For more information, please refer to (Koo et al., 2008).

3 Experiments

In this section, we describe the experiments to evaluate our proposed approach by using CTB4 (Xue et al., 2005) and CDT (Liu et al., 2006). For the former, we adopt a set of head-selection rules (Zhang and Clark, 2008) to convert the phrase structure syntax of treebank into a dependency tree representation. The standard data split of CTB4 from Wang et al. (2007) is used. For the latter, we randomly select 2,000 sentences for test set, another 2,000 sentences for development set, and others for training set.

We use two baseline parsers, one trained on CTB4, and another trained on CDT in the experiments. We choose the n -best size of 16 and the best iteration time of four on the development set since these settings empirically give the best performance. CTB4 and CDT use two different POS tag sets and transforming from one tag set to another is difficult (Niu et al., 2009). To overcome this problem, we use Stanford POS Tagger¹ to train a universal POS tagger on the People’s Daily corpus,² a large-scale Chinese corpus (approximately 300 thousand sentences and 7 million words) annotated with word segmentation and POS tags. Then the POS tagger produces a universal layer of POS tags for both the CTB4 and CDT. Note that the word segmentation standards of these corpora (CTB4, CDT and People’s Daily) slightly differs; however, we do not consider this problem and leave it for future research.

The performance of the parsers is evaluated using the following metrics: UAS, DA, and CM, which are defined by (Hall et al., 2006). All the metrics except CM are calculated as mean scores per word, and punctuation tokens are consistently excluded.

We conduct experiments incrementally to evaluate the joint features used in our first-order and second-order parsers. **The first-order parser**

¹<http://nlp.stanford.edu/software/tagger.shtml>

²<http://www.icl.pku.edu.cn>

-	Features	CTB4		CDT	
		UAS	CM	UAS	CM
dep1	baseline	86.6	42.5	75.4	16.6
	+ <i>edge</i>	88.01 (↑1.41)	44.28 (↑1.78)	77.10 (↑1.70)	17.82 (↑1.22)
	+ <i>root</i>	87.22 (↑0.62)	43.03 (↑0.53)	75.83 (↑0.43)	16.81 (↑0.21)
	+ <i>both</i>	88.19 (↑1.59)	44.54 (↑2.04)	77.16 (↑1.76)	17.90 (↑1.30)
	CTB4 + CDT	87.32	43.08	75.91	16.89
dep2	baseline	88.38	48.81	77.52	19.70
	+ <i>edge</i>	89.17 (↑0.79)	49.73 (↑0.92)	78.44 (↑0.92)	20.85 (↑1.15)
	+ <i>sib</i>	88.94 (↑0.56)	49.26 (↑0.45)	78.02 (↑0.50)	20.13 (↑0.43)
	+ <i>gp</i>	88.90 (↑0.52)	49.11 (↑0.30)	77.97 (↑0.45)	20.06 (↑0.36)
	+ <i>root</i>	88.61 (↑0.23)	48.88 (↑0.07)	77.65 (↑0.13)	19.88 (↑0.18)
	+ <i>all</i>	89.62 (↑1.24)	50.15 (↑1.34)	79.01 (↑1.49)	21.11 (↑1.41)
	CTB4 + CDT	88.91	49.13	78.03	20.12

Table 1: Dependency parsing results on the test set with different joint inference features. Abbreviations: dep1/dep2 = first-order parser and second-order parser; baseline = dep1 without considering any joint inference features; +* = the baseline features conjoined with the joint inference features derived from the heterogeneous treebanks; CTB4 + CDT = we simply concatenate the two corpora and train a dependency parser, and then test on CTB4 and CDT using this single model. Improvements of joint models over baseline models are shown in parentheses.

Type	Systems	≤ 40	Full
D	dep2	90.86	88.38
	MaltParser	87.1	85.8
	Wang et al. (2007)	86.6	-
C	MST _{Malt} †	90.55	88.82
	Martins et al. (2008)†	90.63	88.84
	Surdeanu et al. (2010)†	89.40	86.63
H	Zhao et al. (2009)	88.9	86.1
	<i>Ours</i>	91.48	89.62
S	Yu et al. (2008)	-	87.26
	Chen et al. (2009)	92.34	89.91
	Chen et al. (2012)	-	91.59

Table 2: Comparison of different approach on CTB4 test set using UAS metric. MaltParser = Hall et al. (2006); MST_{Malt}=Nivre and McDonald (2008). Type D = discriminative dependency parsers without using any external resources; C = combined parsers (stacked and ensemble parsers); H = discriminative dependency parsers using external resources derived from heterogeneous treebanks, S = discriminative dependency parsers using external unlabeled data. † The results on CTB4 were not directly reported in these papers, we implemented the experiments in this paper.

(**dep1**) only incorporates head-modifier dependency part (McDonald et al., 2005). **The second-order parser (dep2)** uses the head-modifier and sibling dependency parts (McDonald and Pereira, 2006), as well as the grandparent dependency part (Carreras, 2007; Koo et al., 2008). Table 1 shows the experimental results.

As shown in Table 1, we note that adding more joint inference features incrementally, the dependency parsing performance is improved consis-

tently, for both treebanks (CTB4 or CDT). As a final note, all comparisons between joint models and baseline models in Table 1 are statistically significant.³ Furthermore, we also present a baseline method called “CTB4 + CDT” for comparison. This method first tags both CTB4 and CDT with the universal POS tagger trained on the People’s Daily corpus, then simply concatenates the two corpora and trains a dependency parser, and finally tests on CTB4 and CDT using this single model. The comparisons in Table 1 tell us that very limited information is obtained without consensus features by simply taking a union of the dependencies and their contexts from the two treebanks.

To put our results in perspective, we also compare our second-order joint parser with other best-performing systems. “≤ 40” refers to the sentence with the length up to 40 and “Full” refers to all the sentences in test set. The results are shown in Table 2, our approach significantly outperforms many systems evaluated on this data set. Chen et al. (2009) and Chen et al. (2012) reported a very high accuracy using subtree-based features and dependency language model based features derived from large-scale data. Our systems did not use such knowledge. Moreover, their technique is orthogonal to ours, and we suspect that combining their subtree-based features into our systems might get an even better performance. We do not present the comparison of our proposed approach

³We use the sign test at the sentence level. All the comparisons are significant at $p < 0.05$.

Type	Systems	UAS	DA
D	Duan et al. (2007)	83.88	84.36
	Huang and Sagae (2010)	85.20	85.52
	Zhang and Nivre (2011)	86.0	-
C	Zhang and Clark (2008)	-	86.21
	Bohnet and Kuhn (2012)	87.5	-
H	Li et al. (2012)	86.44	-
	<i>Ours</i>	85.88	86.52
S	Chen et al. (2009)	-	86.70

Table 3: Comparison of different approaches on CTB5 test set. Abbreviations D, C, H and S are as in Table 2.

Treebanks	#Sen	# Better	# NoChange	# Worse
CTB4	355	74	255	26
CDT	2,000	341	1,562	97

Table 4: Statistics on joint inference output on CTB4 and CDT development set.

with the state-of-the-art methods on CDT because there is little work conducted on this treebank.

Some researchers conducted experiments on CTB5 with a different data split: files 1-815 and files 1,001-1,136 for training, files 886-931 and 1,148-1,151 for development, files 816-885 and files 1,137-1,147 for testing. The development and testing sets were also performed using gold-standard assigned POS tags. We report the experimental results on CTB5 test set in Table 4. Our results are better than most systems on this data split, except Zhang and Nivre (2011), Li et al. (2012) and Chen et al. (2009).

3.1 Additional Results

To obtain further information about how dependency parsers benefit from the joint inference, we conduct an initial experiment on CTB4 and CDT. From Table 4, we find that out of 355 sentences on the development set of CTB4, 74 sentences benefit from the joint inference, while 26 sentences suffer from it. For CDT, we also find that out of 2,000 sentences on the development set, 341 sentences benefit from the joint inference, while 97 sentences suffer from it. Although the overall dependency parsing results is improved, joint inference worsens dependency parsing result for some sentences. In order to obtain further information about the error sources, it is necessary to investigate why joint inference gives negative results, we will leave it for future work.

4 Conclusion and Future Work

We proposed a novel framework of joint inference, in which multiple dependency parsing mod-

els were coordinated to search for better dependency parses by leveraging the consensus information between heterogeneous treebanks. Experimental results showed that joint inference significantly outperformed the state-of-the-art baseline models.

There are some ways in which this research could be continued. First, recall that the joint inference scheme involves an iterative algorithm by using bootstrapping. Intuitively, there is a lack of formal guarantee. A natural avenue for further research would be the use of more powerful algorithms that provide certificates of optimality; e.g., dual decomposition that aims to develop decoding algorithms with formal guarantees (Rush et al., 2010). Second, we would like to combine our heterogeneous treebank annotations into a unified representation in order to make dependency parsing results comparable across different annotation guidelines (e.g., Tsarfaty et al. (2011)).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61070106, No. 61272332 and No. 61202329), the National High Technology Development 863 Program of China (No. 2012AA011102), the National Basic Research Program of China (No. 2012CB316300). We thank the anonymous reviewers and the prior reviewers of ACL-2012 and AAI-2013 for their insightful comments. We also thank Dr. Li Cai for providing and preprocessing the data set used in this paper.

References

- B. Bohnet and J. Kuhn. 2012. The best of both worlds—a graph-based completion model for transition-based parsers. In *Proceedings of EACL*.
- X. Carreras. 2007. Experiments with a Higher-order Projective Dependency Parser. In *Proceedings of EMNLP-CoNLL*, pages 957-961.
- W. Chen, D. Kawahara, K. Uchimoto, and Torisawa. 2009. Improving Dependency Parsing with Subtrees from Auto-Parsed Data. In *Proceedings of EMNLP*, pages 570-579.
- W. Chen, M. Zhang, and H. Li. 2012. Utilizing dependency language models for graph-based dependency parsing models. In *Proceedings of ACL*.
- Y. Ding and M. Palmer. 2004. Synchronous dependency insertion grammars: a grammar formalism for syntax based statistical MT. In *Proceedings of*

- the Workshop on Recent Advances in Dependency Grammar*, pages 90-97.
- X. Duan, J. Zhao, and B. Xu. 2007. Probabilistic Models for Action-based Chinese Dependency Parsing. In *Proceedings of ECML/PKDD*.
- J. M. Eisner. 2000. Bilexical Grammars and Their Cubic-Time Parsing Algorithm. Advanced in Probabilistic and Other Parsing Technologies, pages 29-62.
- J. Hall, J. Nivre, and J. Nilsson. 2006. Discriminative Classifier for Deterministic Dependency Parsing. In *Proceedings of ACL*, pages 316-323.
- L. Huang and K. Sagae. 2010. Dynamic Programming for Linear-Time Incremental Parsing. In *Proceedings of ACL*, pages 1077-1086.
- T. Koo, X. Carreras, and M. Collins. 2008. Simple Semi-Supervised Dependency Parsing. In *Proceedings of ACL*.
- T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. 2010. Dual Decomposition for Parsing with Non-Projective Head Automata. In *Proceedings of EMNLP*.
- M. Li, N. Duan, D. Zhang, C.-H. Li, and M. Zhou. 2009. Collaborative Decoding: Partial Hypothesis Re-ranking Using Translation Consensus Between Decoders. In *Proceedings of ACL*, pages 585-592.
- Z. Li, T. Liu, and W. Che. 2012. Exploiting multiple treebanks for parsing with Quasi-synchronous grammars. In *Proceedings of ACL*.
- T. Liu, J. Ma, and S. Li. 2006. Building a Dependency Treebank for Improving Chinese Parser. *Journal of Chinese Languages and Computing*, 16(4):207-224.
- A. F. T. Martins, D. Das, N. A. Smith, and E. P. Xing. 2008. Stacking Dependency Parsers. In *Proceedings of EMNLP*, pages 157-166.
- R. McDonald and F. Pereira. 2006. Online Learning of Approximate Dependency Parsing Algorithms. In *Proceedings of EACL*, pages 81-88.
- R. McDonald, K. Crammer, and F. Pereira. 2005. Online Large-margin Training of Dependency Parsers. In *Proceedings of ACL*, pages 91-98.
- Z. Niu, H. Wang, and H. Wu. 2009. Exploiting Heterogeneous Treebanks for Parsing. In *Proceedings of ACL*, pages 46-54.
- J. Nivre and R. McDonld. 2008. Integrating Graph-based and Transition-based Dependency Parsing. In *Proceedings of ACL*, pages 950-958.
- A. M. Rush, D. Sontag, M. Collins, and T. Jaakkola. 2010. On Dual Decomposition and Linear Programming Relation for Natural Language Processing. In *Proceedings of EMNLP*.
- M. Surdeanu and C. D. Manning. 2010. Ensemble Models for Dependency Parsing: Cheap and Good? In *Proceedings of NAACL*.
- R. Tsarfaty, J. Nivre, and E. Andersson. 2011. Evaluating Dependency Parsing: Robust and Heuristics-Free Cross-Annotation Evaluation. In *Proceedings of EMNLP*.
- J.-N Wang, J.-S. Chang, and K.-Y. Su. 1994. An Automatic Treebank Conversion Algorithm for Corpus Sharing. In *Proceedings of ACL*, pages 248-254.
- Q. I. Wang, D. Lin, and D. Schuurmans. 2007. Simple Training of Dependency Parsers via Structured Boosting. In *Proceedings of IJCAI*, pages 1756-1762.
- N. Xue, F. Xia, F.-D. Chiou, and M. Palmer. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 10(4):1-30.
- Yamada and Matsumoto. 2003. Statistical Sependency Analysis with Support Vector Machines. In *Proceedings of IWPT*, pages 195-206.
- D. H. Younger. 1967. Recognition and Parsing of Context-Free Languages in Time n^3 . *Information and Control*, 12(4):361-379, 1967.
- K. Yu, D. Kawahara, and S. Kurohashi. 2008. Chinese Dependency Parsing with Large Scale Automatically Constructed Case Structures. In *Proceedings of COLING*, pages 1049-1056.
- Y. Zhang and S. Clark. 2008. A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing Using Beam-Search. In *Proceedings of EMNLP*, pages 562-571.
- Y. Zhang and J. Nivre. 2011. Transition-based Dependency Parsing with Rich Non-local Features. In *Proceedings of ACL*, pages 188-193.
- H. Zhao, Y. Song, C. Kit, and G. Zhou. 2009. Cross Language Dependency Parsing Using a Bilingual Lexicon. In *Proceedings of ACL*, pages 55-63.
- G. Zhou, L. Cai, J. Zhao, and K. Liu. 2011. Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives. In *Proceedings of ACL*, pages 653-662.
- G. Zhou, J. Zhao, K. Liu, and L. Cai. 2011. Exploiting Web-Derived Selectional Preference to Improve Statistical Dependency Parsing. In *Proceedings of ACL*, pages 1556-1565.
- G. Zhou, L. Cai, K. Liu, and J. Zhao. 2011. Improving Dependency Parsing with Fined-Grained Features. In *Proceedings of IJCNLP*, pages 228-236.
- M. Zhu, J. Zhu, and T. Xiao. 2010. Heterogeneous Parsing via Collaborative Decoding. In *Proceedings of COLING*, pages 1344-1352.