Improving Chinese Dependency Parsing with Self-Disambiguating Patterns

Likun Qiu[1,3], Lei Wu[2,3], Kai Zhao[3], Changjian Hu[3], Lingpeng Kong[3]
[1] Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, Beijing, China
[2]Institute of Automation, China Academy of Science
[3]NEC Laboratories, Beijing, China
qiulikun@pku.edu.cn

***Abstract*—To solve the data sparseness problem in dependency parsing, most previous studies used features constructed from large-scale auto-parsed data. Unlike previous work, we propose a new approach to improve dependency parsing with context-free dependency triples (CDT) extracted by using self-disambiguating patterns (SDP). The use of SDP makes it possible to avoid the dependency on a baseline parser and explore the influence of different types of substructures one by one. Additionally, taking the available CDTs as seeds, a label propagation process is used to tag a large number of unlabeled word pairs as CDTs. Experiments show that, when CDT features are integrated into a maximum spanning tree (MST) dependency parser, the new parser improves significantly over the baseline MST parser. Comparative results also show that CDTs with dependency relation labels perform much better than CDT without dependency relation label.**

Dependency parsing; self-disambiguating pattern; raw corpus (key words)

I. INTRODUCTION

To obtain dependency parsers with high accuracy, one promising direction is to use knowledge acquired from large-scale unannotated text, e.g., substructures extracted from auto-parsed data (i.e. "verb-object" and "modifier-head" structures in [1], case structure in [2] and subtrees in [3]).

However, since most of the substructures are extracted based on auto-parsed results, and the accuracies of auto-parsing is not high (about 82% [2] on real POS-tag), there are many errors in the extracted substructures. Naturally, these errors might decrease the performance of dependency parsing. Moreover, most of previous researches (except [1]) used all kinds of substructures and took them as one type[1]. So it is difficult for us to know the influence of each type of substructures and hard to achieve further improvements.

Instead of extracting from auto-parsed data, we propose an approach to extract substructures directly from auto-segmented and auto-POS-tagged data. The approach is referred to as SDP-based approach. Here, SDP denotes self-disambiguating pattern, which could resolve the ambiguity of a syntactic structure by itself. Since the precision of word segmentation and POS-tagging is much higher[2] than syntactic parsing and SDP could resolve syntactic ambiguity in a certain degree, the SDP-based method could extract labeled substructures effectively.

Moreover, unlike previous studies which improved performance by either using only verb-noun relations [1], or considering all kinds of syntactic collocations as only one pattern [2-3], we propose to view substructures differently according to the dependency relations.

To demonstrate the effectiveness of the proposed approach, we made experiments on Penn Chinese Tree Bank (CTB). The results showed that the proposed approach greatly improves the accuracy over the baseline parser and outperforms the state-of-art system. We also demonstrate that the label propagation process further improve the dependency parsing by tagging more CDTs.

The rest of this paper is organized as follows. Section 2 introduces how to extract CDTs by using SDPs; Section 3 proposes label propagation method; Section 4 explains Chinese dependency parser using dependency triples. Experimental results are showed in section 5. Finally, a brief conclusion and our future work plan will be given in Section 6.

II. SDP-BASED APPROACH FOR CDT EXTRACTION

In this paper, we would use SDPs to extract CDTs from large scale unannotated data.

A. CDT

A CDT is a context-free dependency triple with the form of $\{w_1, r, w_2\}$, in which r is the dependency relation between two words w_1 and w_2. In the sentence in Fig. 1, the word 解决 (solve) is in four CDTs: {村子 (village), SUB, 解决 (solve)}, {已经 (already), VMOD, 解决 (solve)}, {解决 (solve), VMOD, 了 (le)}, {解决 (solve), OBJ, 问题 (problem)}. After extracted from many different sentences, the relation type of a dependency triple is not dependent on its contexts. Therefore, we call it context-free.

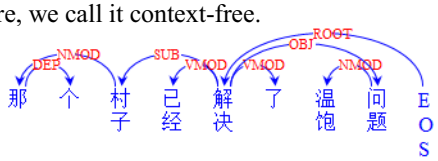

Figure 1. Example of Dependency Tree

B. SDP

There exist many ambiguous syntactic structures in natural languages. For instance, in Chinese, the dependency relation between a verb and a noun might be OBJ or NMOD; the relation between two verbs might be OBJ, NMOD, or VMOD.

[1] Here, "type" denotes syntactic relation types such as coordinate, predicate-object, etc.
[2] Currently, the best performance of Chinese word segmentation has achieved 99.20% on F-score, and the best accuracy of Chinese POS-tagging was 96.89% [4].

978-0-7695-4554-7/11 $26.00 © 2011 IEEE
DOI 10.1109/IALP.2011.36

However, in some cases, the relation of words can be decided without uncertainty. For example, although relation between a verb and a noun might be OBJ or NMOD, but in the case: "Verb + 了 + Numeral + Quantifier + Anywords + Noun + EOS"[3], the relation of the verb and the noun is almost certainly OBJ. For example, in "收到(receive)+了+一(a)+束(bunch)+漂亮的(beautiful)+鲜花(flower)+EOS", the relation between "收到" and "鲜花" is OBJ. Such pattern is referred to as self-disambiguating pattern (SDP).

C. SDP-based Approach for CDT Extraction

TABLE I. INSTANCES OF CDTS

Verb	Noun	Frequency
解决(solve)	问题(problem)	64965
提出(raise)	要求(claim)	54313
发挥(play)	作用(role)	46052
取得(score)	成绩(achievement)	42867
奠定(lay)	基础(foundation)	37495

TABLE II. SDPS USED IN THIS PAPER

Target CDT	SDP
<Noun, SUB, Verb>	BOS + [Adjective | Noun | Pronoun | Numeral | Quantifier |的]* + Noun + Adverb + Verb + 了 + [Adjective | Noun | Pronoun| Numeral | Quantifier |的]* + Noun + EOS
<Verb, OBJ, Noun>	Verb + 了 + Numeral + Quantifier + Anywords + Noun + EOS
<Quantifier, NMOD, Noun>	了 + Numeral + NominalQuantifier + Anywords + Noun + EOS
<Noun, NMOD, Verb>	的 + Noun + Verb + EOS
<Verb, NMOD, Noun>	了 + Numeral + NominalQuantifier + Anywords +Verb + Noun + EOS
	的 + Verb + Noun + EOS

Since SDP can resolve the ambiguity of a structure, we can use it to extract correct CDTs. For instance, given a SDP such as "*Verb + 了+ Numeral + Quantifier + Anywords + Noun + EOS*", we might extract many instances of <Verb, OBJ, Noun>. Some high-frequency CDTs are show in Table I.

For each ambiguous structure, we might design one or more SDPs. Therefore, the number of SDPs in a certain language might be very large. In this paper, we present five SDPs to resolve three ambiguous structures (see Table II) in Chinese. In this table, the first column denotes the target CDT that could be extracted by the SDP in the second column. The SUB and OBJ structure form the stretch of a common sentence; the noun-verb NMOD structure and verb-noun NMOD structure are the corresponding ambiguous structure of SUB and OBJ, respectively; the quantifier-noun NMOD structure is a typical relation of Chinese, where the quantifier and noun select mutually. That's why we choose them in our experiment.

[3] Here, "EOS" denotes the end of a sentence or clause; "Anywords" denotes any words occurring zero or many times; "[*w*]*" means the word *w* occurs zero or many times.

III. CHINESE DEPENDENCY PARSER USING CDT FEATURES

We generate new features based on the extracted CDTs and refer them as CDT-based features. Since these features only contain two words, they correspond to the first-order features in the MST parsing model. Second-order or higher-order features would not be tried in this paper.

A CDT-based feature is represented as follows:

$$feature(w_i,w_j)=id_{i,j}\text{-}type_{i,j}\text{-}direction_{i,j}$$

where $id_{i,j}$, $type_{i,j}$ and $direction_{i,j}$ denote the frequency, dependency relation type and dependency direction of the CDT respectively.

All the extracted CDTs are grouped into different sets in terms of frequencies. With experiments and reference to [3], we chose the following way. CDTs are grouped into three sets: "high-frequency (HF)", "middle-frequency (MF)" and "low-frequency (LF)". HF, MF and LF are used as set IDs for the three sets respectively. The following are the settings: if the frequency of a CDT is larger than the threshold λ_1, it is in set HF; else if the frequency is larger than λ_2, it is in set MF; else it is in set LF. We store the set ID for every CDT in L_{st}.

The dependency type set contains three elements: SUB, OBJ and NMOD. The dependency directions of SUB, OBJ and NMOD are "left", "right" and "left" respectively. Here, "left" means in the CDT the left word depends on the right word. If a CDT with SUB label is matched, the value of "direction" would be set as "left".

For instance, if the frequency of <解决 (solve), OBJ, 问题 (problem)> is larger than λ_1, its set ID is HF. Then the system would generate a feature of HF-OBJ-Right.

IV. CDT EXPANSION : LABEL PROPAGATION PROCESS

Since large scale unannotated data is also limit, the data sparseness problem still exists. To solve this problem further, we try to use a Label Propagation process to tag dependency pairs as CDTs.

The problem of CDT Tagging is to assign an appropriate dependency relation type to a context-free dependency pair. It can be represented as follows:

$$r->(w_i,w_j)$$

where w_i and w_j denote the first and second word of a dependency pair. The CDT tagging problem is to assign an appropriate dependency type r to the pair (w_i, w_j). Then, we would get a CDT < w_i, r, w_j >.

To solve the CDT tagging problem, we need to do three steps. The first is to acquire large scale dependency pairs. The second is to measure the similarity between two dependency pairs. And the third is to apply the label propagation algorithm.

For the sub-problem of dependency pair acquisition, we directly use the result of [5][4], which used several statistical methods to extract 18M word pairs on a data set containing about 100M web pages.

For word similarity, we used the method proposed by (Curran, 2006) to compute Chinese word similarity. Given a corpus and a set of words CS(w). Collect context of the words in CS(w) from a corpus. Denote context of a headword w_1 as CT(w_1), where w1∈ CS(w). The context of a headword w_1 includes 2 words before w_1 and 2 words after w_1. Define contextual similarity of w_i and w_j as Sim(w_i, w_j). Cosine distance is used to compute Sim(w_i, w_j). See the following formula for concrete definition, where n denotes the dimension of the two vectors, while v_{ik} and v_{jk} denote the weighted value of the kth context word of w_i and w_j respectively.

$$Sim(w_i, w_j) = \frac{\sum_{j=1}^{n} v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^{n} v_{ik}^2} \sqrt{\sum_{k=1}^{n} v_{jk}^2}}$$

As for the label propagation algorithm, we follow [7].

V. EXPERIMENTS

A. Experimental Setting

We made experiments on the Penn Chinese Treebank (CTB)5.0 [5] [8]. CTB is converted into dependency structures using a standard set of head rules by the tool "Penn2Malt" [6] (Yamada and Matsumoto, 2003). In CTB 5.0, Section 1-270 and 400-931 are used for training, Section 271-300 for testing, and Section 301-325 for development. Data partition and POS-tags on CTB 5.0 are the same as the settings in [2-3, 9]. All the evaluation metrics are calculated on the dependency relations in which the modifier is not a punctuation.

Two unannotated corpus are used for extracting CDTs. The first one is a self-made corpus, called Raw-Corpus. Raw-Corpus contains about 20M sentences, which are collected from Chinese news websites from January to December 2006. We use the second order MSTParser[7] as our baseline parser. It is trained with Section 1-270 and 400-931 of CTB 5.0. All the sentences in the Raw-Corpus are parsed by the baseline parser for extracting CDTs from auto-parsed sentences directly.

The second corpus is Sogou Web Corpus[8] V2.0. This corpus contains 120G web pages, yet we only used the previous 30G for extracting CDTs by SDP-based method.

We use the same feature with [10] and adopt the default settings of MSTParser throughout the paper: iters=10; training-k=1; decode-type=proj; order=2. When grouping CDTs by frequencies, the parameters λ_1 and λ_2 are set as 100 and 10 respectively.

B. Experimental Results

Results of CDT Extraction

The results of CDT extraction are show in Table III. There are 55,175 and 1485 dependency triples (SUB, OBJ, NMOD) in training set and test set respectively. Auto-parsed CDTs are extracted from auto-parsed result (using the baseline parser) on the Chinese Raw-Corpus. SDP-based CDTs are extracted from Sogou Corpus based on the five proposed SDPs and using ICTCLAS2009 (Zhang, 2002)[9] as the segmentation and POS-tagging tool. The proposed result is expanded by the label propagation process based on the SDP-based CDTs.

Since the dependency triples in training set and test set are context-dependent yet CDT is context-free, it is difficult for us to evaluate CDT in terms of precision. We only evaluate the coverage rate of CDT, i.e. the percentage of CDTs in training set and test set that have been covered by the given CDT set. Table III shows that auto-parsed CDTs can cover more than SDP-based CDTs. It is mainly because we only used five SDPs and can only cover a small part of the SUB, OBJ and NMOD CDTs. The usefulness of the three CDT sets would be evaluated by dependency parsing evaluation.

TABLE III. CDT QUANTITY

CDT Source	Quantity	Coverage Rate
Training Set	55,175	-
Test Set	1485	-
Auto-parsed	8.52M	53.9
SDP-based	1.69M	16.1
Proposed	3.09M	22.2

5.2.2 Results of Dependency Parsing

The dependency parsing results of proposed parsers are show in Table IV. Five parsers are compared together with a baseline parser:

- Baseline: We use the second order MSTParser as our baseline parser.
- Auto1: The parser which uses CDTs extracted from auto-parsed data, without dependency label.
- Auto2: The parser which uses CDTs extracted from auto-parsed data, with dependency label.
- SDP1: The parser which only uses seed CDTs without dependency label. That is, all kinds of CDT are considered as one kind.
- SDP2: The parser which uses seed CDTs with different dependency label.
- Proposed: The proposed parser, in which both seed CDTs and expanded CDTs are used.

Note that all the SDP1/2 and Auto1/2 parsers only used CDTs of SBJ, OBJ and NMOD relations.

Table IV shows that all the five parsers outperform the baseline parsers. There is an absolute improvement of 1.26 points (UAS) by adding CDT-based features in the proposed parser. The improvement of parsing with CDT-based features is significant in McNemar's Test ($p < 10^{-5}$). Fig. 3 and Fig. 4 show the dependency trees of the same sentences created by the baseline parser and the proposed parser, respectively. After using the CDT <会谈 (interview), SUB, 具有 (have)>, the correct subject of 具有 (have) was found by the proposed parser.

[4] The collocation set might be downloaded from http://www.sogou.com/labs/dl/r.html.
[5] http://www.cis.upenn.edu/~chinese/
[6] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html
[7] http://mstparser.sourceforge.net
[8] http://www.sogou.com/labs/dl/t.html
[9] http://ictclas.org/index.html

The comparative results between SDP/Auto1 and SDP/Auto2 parsers show that integrating dependency relation type into features is very useful for dependency parsing.

The comparative results between SDP1/2 and Auto1/2 parser show that SDP-based method could extract CDTs effectively. Note that SDP1/2 parser, which used five SDPs and only cover about 16% dependency triples in CTB 5.0, outperform the parser which used all CDTs from auto-parsed data and cover 53.9% dependency triples.

We also checked the effect of expanding CDTs by label propagation process. The comparative results between SDP2 parser and proposed parser show that label propagation process can provide further improvement, although not very significantly.

TABLE IV. DEPENDENCY PARSING RESULTS FOR PROPOSED PARSERS

Parser	UAS(%)	LAS(%)
Baseline	88.21	87.19
Auto1	88.71 (+0.50)	87.71(+0.52)
SDP1	88.72 (+0.51)	87.71(+0.52)
Auto2	89.19 (+0.98)	88.14(+0.95)
SDP2	89.25 (+**1.04**)	88.21(+**1.02**)
Proposed	89.47(+**1.26**)	88.43(+**1.24**)

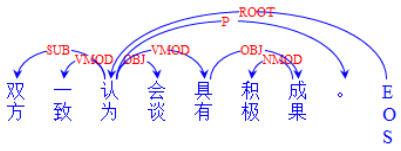

Figure 2. Result created by the baseline parser

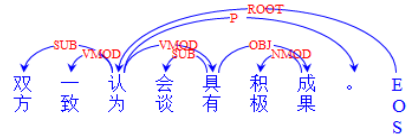

Figure 3. Result created by the proposed parser

TABLE V. DEPENDENCY PARSING RESULTS FOR THE PROPOSED PARSERS AND PREVIOUS WORK

Parser	UAS(%)	LAS(%)
Baseline	88.21	87.19
Chen08	86.52	-
Yu08	87.26	-
Chen092b	89.16	-
Chen092s	89.43	-
Chen10	89.43	
Proposed	89.47	88.43

5.2.3 Comparative Results of Dependency Parsing

Table V shows the comparative results, where Chen08 refers to the parser of [8], Yu08 refers to the parser of [2], Chen09b/s refers to the parsers of [3] and Chen10 refers to the parsers of [11]. Specially, Chen09b only used bigram features yet chen09s used both bigram and trigram features. [3] and [11] achieve UAS score of 89.91% and 89.53% respectively, yet both the two scores are achieved by combining their algorithm with other kind of method together.

The results show that our proposed parser not only outperforms previous best results (not including results of hybrid methods) that used bigram features, but also perform a little better than the results achieved by employing both bigram and trigram features.

VI. CONCLUSION AND FUTURE WORK

We present an effective approach to improve dependency parsing using CDTs extracted by SDP-based method and a label propagation process. The experiments show that the SDP-based method can improve dependency parsing significantly. Experimental results also show that CDTs with dependency labels perform much better than that without dependency label.

Much work can be done to further exploit SDP-based method. For example, we only used five SDPs in this paper, which can only cover a small part of the dependency triples. We would design more SDPs to deal with dependency relations such as adverbial-head, predicate-complement and coordinate structure.

ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the China Postdoctoral Science Foundation.

REFERENCES

[1] A. Wu. 2003. Learning Verb-Noun Relations to Improve Parsing. In Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing. pages.119-124.

[2] K. Yu, D. Kawahara, and S. Kurohashi. 2008. Chinese Dependency Parsing with Large Scale Automatically Constructed Case Structures. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1049-1056. Manchester, August 2008.

[3] W. Chen, J. Kazama, K. Uchimoto, and K. Torisawa. 2009. Improving Dependence Parsing with Subtrees from Auto-Parsed Data. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 570-579. Singapore, 6-7 August 2009.

[4] G. Jin and X. Chen. 2008. The Fourth International Chinese Language Processing Bakeoff: Chinese Word Segmentation, Named Entity Recognition and Chinese Pos Tagging. In Proceedings of the 6th SIGHAN Workshop on Chinese Language Processing, pages 69-81.

[5] D. Wang, X. Tu, X. Zheng and Z. Tong. 2008. Collocation Extraction with Multiple Hybrid Strategies. Journal of Tsinghua University (Science & Technology), Vol. 48, No. 4, pages 608-612.

[6] J. R. Curran. 2005. Supersense Tagging of Unknown Nouns using Semantic Similarity. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 26-33.

[7] X. Zhu and Z. Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. CMU CALD tech report CMU-CALD-02-107.

[8] W. Chen, D. Kawahara, K. Uchimoto, Y. Zhang, and H. Ishara. 2008. Dependency parsing with short dependency relations in unlabeled data. In Proceedings of IJCNLP 2008.

[9] R. McDonald, and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In Proceedings of EACL 2006.

[10] N. Xue, F. Chiou, and M. Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In Proceedings of Coling 2002.

[11] W. Chen, J. Kazama, Y. Tsuruoka, and K. Torisawa. 2010. Improving Graph-based Dependency Parsing with Decision History. In Proceedings of Coling 2010.