Learning Head-modifier Pairs to Improve Lexicalized Dependency Parsing on a Chinese Treebank

Kun Yu
Kyoto University
Graduate School of Informatics

Daisuke Kawahara
National Institute of Information and Communications Technology
Knowledge Creating Communication Research Center

Sadao Kurohashi
Kyoto University
Graduate School of Informatics

Abstract

Due to the data sparseness problem, the lexical information from a treebank for a lexicalized parser could be insufficient. This paper proposes an approach to learn head-modifier pairs from a raw corpus, and to integrate them into a lexicalized dependency parser to parse a Chinese Treebank. Experimental results show that this approach not only enlarged the coverage of bi-lexical dependency, but also improved the accuracy of dependency parsing significantly.

1 Introduction

The emergence of treebank of many languages makes it possible to use lexical information in parsers, such as Collins (1999) and Charniak (2000) for English, Uchimoto et al. (2000) and Kudo and Matsumoto (2002) for Japanese, and Cao et al. (2005) for Chinese. But, due to the data sparseness problem, the state-of-the-art lexicalized parsers are mainly based on un-lexical information. For example, Bikel (2004) indicated that Collins's parser used bi-lexical dependencies only 1.49% of the time. In other cases, it backed off to condition a word on its phrasal and part-of-speech category.

Head-modifier pairs, which mean pos-tagged word pairs with dependency relations, can help recognize lexical preference for parsing. For example, in the Chinese sentence shown in Figure 1, the head-modifier pair '浦东/NN→建设/NN'[1] can help recognize the correct head of '浦东/NN' as '建设/NN' when there only exists back-off dependency 'NN→NN' [2] in training data.

---

[1] The head-modifier pair 'w_k/p_k→w_h/p_h' means word w_h with pos-tag p_h is the head of word w_k with pos-tag p_k. All the pos-tags appearing in this paper follow the definition in Penn Chinese Treebank.

[2] The dependency 'p_k→p_h' means any word with pos-tag p_h is head of any word with pos-tag p_k. It is the back-off of dependency 'w_k/p_k→w_h/p_h', which means word w_h with pos-tag p_h is the head of word w_k with pos-tag p_k.

De Smedt, K., Hajič, J. and Kübler, S. (Eds.)
Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (2007)

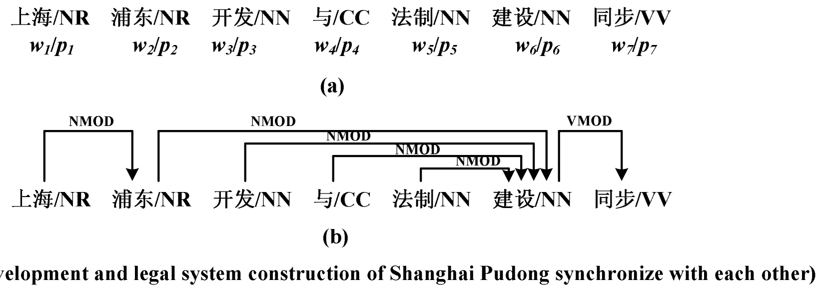

Figure 1. A Chinese sentence and its dependency tree.
(a) input sentence with word and pos-tag; (b) dependency tree of input sentence

In this paper, we proposed an approach which learns head-modifier pairs automatically from a large raw corpus to introduce more bi-lexical dependencies that could not be obtained from Treebank due to data sparseness problem, and then uses these head-modifier pairs to improve lexicalized dependency parsing on a Chinese Treebank. In the proposed approach, the raw corpus is first segmented and pos-tagged by an existing morphological analyzer, and then it is parsed by a deterministic parser. Finally, reliable head-modifier pairs are extracted from the parsed sentences and their probabilities are calculated.

We did experiments on Penn Chinese Treebank 5.1 (Xue et al., 2002). The experimental results proved that by using the learned head-modifier pairs, not only the coverage of bi-lexical dependency was enlarged by 29.55%, but also the dependency accuracy of the lexicalized dependency parser was increased by 0.61% significantly.

The rest of the paper is organized as follows. Section 2 shows the way to learn head-modifier pairs from a raw corpus. A lexicalized dependency parser used as test bed is described briefly in Section 3. Section 4 shows how to integrate the learned head-modifier pairs into this parser. Experimental results are discussed in Section 5. Section 6 introduces the related work. Finally, Section 7 gives a brief conclusion and indicates the directions for the future work.

2 Learning Head-modifier Pairs from Chinese Gigaword

2.1 Preprocessing of Chinese Gigaword

We choose Chinese Gigaword (Graff et al., 2005) as the raw corpus to learn head-modifier pairs. In this corpus, 1,033,679 files written in simplified Chinese are used, which include 499,176,000 Chinese characters totally.

The raw corpus is first segmented and pos-tagged by a Chinese morphological analyzer (Nakagawa and Uchimoto, 2007). Then a Chinese deterministic parser (Yu et al., 2007) is applied to parse the whole corpus.

ISSN 1736-6305 Vol. 1
http://dspace.utlib.ee/dspace
/handle/10062/4476

2.2 Extracting Reliable Head-modifier Pairs

To calculate probability of head-modifier pairs precisely, we need to extract reliable head-modifier pairs from the parsed corpus. Whether a head-modifier pair is reliable depends on the parsing accuracy. Thus we first select good parses from all the parsed sentences, and then extract reliable head-modifier pairs from these good parses.

➢ **Selecting good parses**

In the proposed approach, we look sentence as the unit for good parse selection, and assume that the parse of a short sentence is more accurate than the parse of a long sentence. Figure 2 shows the different dependency accuracy (see equation 10) of the deterministic parser on 1,800 sentences from Penn Chinese Treebank 5.1 with different *maximum sentence length*[3]. It is obvious that the dependency accuracy increases while the maximum sentence length decreases. Thus, we choose maximum sentence length as a criterion for good parse selection.

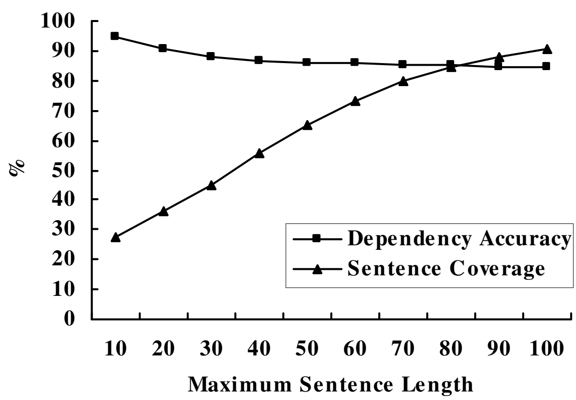

Figure 2. Dependency accuracy and sentence coverage on different maximum sentence length (gold word segmentation and pos-tag are used in the experiment)

Table 1. Dependency Accuracy of the deterministic parser on sentences with no more than 30 words (%)

Dependency Type[4]	all	[N,V]	[V,V]	[N,N]	[V,P]	other
Dependency Accuracy	88.14	90.17	65.08	88.03	85.01	91.04

But, from Figure 2 we also find the sentence coverage decreases quickly together with the decrease of maximum sentence length. Therefore, we need to find a trade-off between dependency accuracy and sentence coverage. In the

[3] *maximum sentence length* means the maximum number of words in one sentence
[4] 'all, [N,V], [V,V], [N,N], [V, P], other' are different dependency types which will be introduced in Section 5.3.

De Smedt, K., Hajič, J. and Kübler, S. (Eds.)
Proceedings of the Sixth International Workshop
on Treebanks and Linguistic Theories (2007)

proposed approach, we choose the threshold of maximum sentence length as 30 words empirically and extract all the sentences whose length is no more than this threshold as good parses. By this way, 44.95% (3,298,198) sentences from Chinese Gigaword are selected, and the detailed dependency accuracy of the deterministic parser is shown in Table 1.

- **Extracting head-modifier pairs from good parses**

After selecting the good parses, we look all the head-modifier pairs in the selected sentences as reliable pairs and extract them. Totally, we get 1,368,232 reliable head-modifier pairs.

2.3 Calculating Probability for Head-modifier Pairs

The probability of head-modifier pairs, which we call as P_{HM}, represents the probability of one word to be modifier given the other word. The maximum likelihood estimation of this probability is shown in equation 1.

$$\hat{P}_{HM}(w_k / p_k \mid w_h / p_h) = \frac{count(w_k / p_k \rightarrow w_h / p_h)}{\sum_i count(w_i / p_i \rightarrow w_h / p_h)} \quad (1)$$

Here w_k/p_k represents a word w_k with pos-tag p_k. $count(w_k / p_k \rightarrow w_h / p_h)$ indicates the number of head-modifier pairs in which w_h/p_h is head of w_k/p_k. $\sum_i count(w_i / p_i \rightarrow w_h / p_h)$ means the number of all the head-modifier pairs in which w_h/p_h is head.

3 A Lexicalized Dependency Parser for Parsing Penn Chinese Treebank

We developed a Chinese lexicalized dependency parser as test bed. As one of the most famous parsing models, Collins's statistical dependency parsing model (Collins, 1996) was selected as the basic model of our dependency parser. The aim of this parser is to take a pos-tagged sentence $S=<w_1/p_1, w_2/p_2,\ldots, w_n/p_n>$ (see Figure 1(a)) as input and then create a dependency tree T_{best} (see Figure 1(b)) as output. CKY algorithm is applied to decode the parse tree from bottom to up.

$$T_{best} = \arg\max_T P(T \mid S) \quad (2)$$

In our lexicalized dependency parser, a Chinese sentence is represented as the combination of a set of **baseNPs (B)**, a set of **conjunctive structures (C)**, and a set of **dependencies (D)** (see Figure 3). Thus $T=(B,C,D)$ and

$$P(T \mid S) = P(B, C, D \mid S) = P(B \mid S) \times P(C \mid B, S) \times P(D \mid S, B, C) \quad (3)$$

ISSN 1736-6305 Vol. 1
http://dspace.utlib.ee/dspace
/handle/10062/4476

B： {上海/NR　浦东/NR}

C： {开发/NN　与/CC　法制/NN　建设/NN}

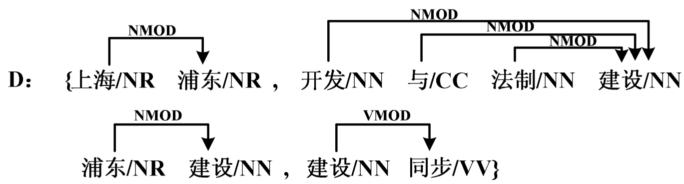

Figure 3. Representation of the Chinese sentence in Figure 1

In equation 3, the baseNP model $P(B|S)$ is estimated by

$$P(B \mid S) = P(t_1 t_2 ... t_n \mid w_1 / p_1, w_2 / p_2, ... w_n / p_n) = \prod_{i=1}^{n} P(t_i \mid w_i / p_i) \qquad (4)$$

where t_i is the baseNP tag for w_i/p_i (Yu et al., 2006). IOB tag definition is used for this tagging process.

The conjunctive structure model $P(C|B,S)$ is estimated by the string-similarity method proposed in Kurohashi and Nagao (1994). A Chinese thesaurus HowNet[5] is used for the similarity calculation between words.

$P(D|S,B,C)$ is estimated by a dependency version of Penn Chinese Treebank. The phrase structure of Penn Chinese Treebank is transferred into dependency structure with dependency label by the toolkit Penn2Malt[6].

To estimate $P(D|S,B,C)$, dependency D is first represented as $D=\{D_k|1 \leq k \leq m\}$, supposing there are m dependency relations in D totally. D_k is a triple represented as $D_k = (w_k/p_k, w_h/p_h, R_k)$, which means w_k/p_k modifies w_h/p_h with dependency label R_k, such as VMOD and NMOD in Figure 1(b). Then we can get

$$P(D \mid S, B, C) = \prod_{k=1}^{m} P(D_k \mid S, B, C) \qquad (5)$$

Referring to Collins's model (Collins, 1996), if we define $P(R_k| w_k/p_k, w_h/p_h)$ as the probability that w_k/p_k modifies w_h/p_h with dependency label R_k, the maximum-likelihood estimate of $P(R_k| w_k/p_k, w_h/p_h)$ is

$$\hat{P}(R_k \mid w_k / p_k, w_h / p_h) = \frac{count(w_k / p_k \xrightarrow{R_k} w_h / p_h)}{count(w_k / p_k, w_h / p_h)} \qquad (6)$$

Here $count(w_k / p_k \xrightarrow{R_k} w_h / p_h)$ is the number of times that w_k/p_k modifies w_h/p_h with dependency label R_k, and $count(w_k / p_k, w_h / p_h)$ is the number of times that w_k/p_k and w_h/p_h co-occur in the sentence.

Therefore, we can get

[5] http://www.keenage.com
[6] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

De Smedt, K., Hajič, J. and Kübler, S. (Eds.)
Proceedings of the Sixth International Workshop
on Treebanks and Linguistic Theories (2007)

$$P(D_k \mid S,B,C) \approx \hat{P}(R_k \mid w_k / p_k, w_h / p_h) \quad (7)$$

To handle the data sparseness problem, the back-off estimation strategy used in Collins (1996) is applied in this parser.

4 Integrating Head-modifier Pairs into Lexicalized Dependency Parsing

To apply the head-modifier pairs into the lexicalized dependency parser introduced in Section 3, we change the definition of $P(D_k|S,B,C)$ to be

$$P(D_k \mid S,B,C) \approx \hat{P}(R_k \mid w_k / p_k, w_h / p_h) \times \hat{P}_{HM}(w_k / p_k \mid w_h / p_h) \quad (8)$$

By this way, the probability of head-modifier pairs is combined together with the probability of dependencies estimated by a treebank.

5 Experimental Results and Discussion

5.1 Data Set

We use Penn Chinese Treebank 5.1 as data set in the experiments. The toolkit Penn2Malt is applied to transfer the phrase structure to dependency structure. 9,684 sentences from Section 001-270, 400-931 are used to estimate $P(R_k| w_k/p_k, w_h/p_h)$ by equation 6. 346 sentences from Section 271-300 are used as testing data. All the sentences in Section 1-9 are used to train the deterministic parser used in Section 2.1. Because our intuition of the experiments is to prove the effectiveness of applying head-modifier pairs into statistical dependency parsing, gold standard word segmentation and pos-tag are used in all the experiments.

5.2 Results of Bi-lexical Dependency Coverage

The objective of using head-modifier pairs in lexicalized dependency parsing is to introduce lexicalized preferences into parser, which could not be obtained from a treebank due to data sparseness problem. Thus we first compare the coverage of bi-lexical dependency (see equation 9) on the gold standard data set for both head-modifier pairs learned from Chinese Gigaword and the dependencies learned from Penn Chinese Treebank.

$$Bilex.Cov. = \frac{\text{\# of bilexical dependency existing in gold standard}}{\text{\# of all bilexical dependency in gold standard}} \quad (9)$$

Table 2 shows that compared with the dependencies learned from Penn Chinese Treebank, the head-modifier pairs made the bi-lexical dependency coverage increase from 41.52% to 71.07%. This result indicates that the learned head-

ISSN 1736-6305 Vol. 1
http://dspace.utlib.ee/dspace
/handle/10062/4476

modifier pairs can introduce bi-lexical dependencies for the lexicalized dependency parser successfully.

Table 2. Coverage of Bi-lexical Dependency

	Bilex.Cov. (%)
Dependency from Treebank	41.52
Head-modifier pair	71.07(+29.55)

5.3 Results of Lexicalized Dependency Parsing

Previous experiment proved that the learned head-modifier pairs can introduce bi-lexical dependencies successfully. In this experiment, we would like to verify that the accuracy of the lexicalized dependency parser can also be improved when parsing a treebank by using the bi-lexical dependencies.

Two models of the parser introduced in Section 3 are tested in this experiment.

- **'w/o'** is the baseline model, which estimates $P(D_k|S,B,C)$ by equation 7
- **'w/HM'** is the proposed approach, which estimates $P(D_k|S,B,C)$ by equation 8

We choose dependency accuracy (see equation 10), which is widely used for evaluating dependency parser, as the main evaluation metric in this experiment. Besides, dependency coverage (see equation 11) is calculated as an auxiliary evaluation metric.

$$Dpnd.Accu = \frac{\#\text{ of correct detected dependency}}{\#\text{ of detected dependency}} \tag{10}$$

$$Dpnd.Cov = \frac{\#\text{ of correct detected dependency}}{\#\text{ of gold standard dependency}} \tag{11}$$

We also classify the dependencies into five types to analyze the results in detail. '[N,V]' means head is verb and modifier is noun; '[V,V]' means both head and modifier are verb; '[N,N]' means both head and modifier are noun; '[V,P]' means head is verb and modifier is preposition or vice versa; 'other' means other types of dependencies, such as head is noun and modifier is adjective. 'all' means all types of dependencies.

Table 3 and Table 4 list the experimental results. These results show that through using the bi-lexical dependencies introduced by head-modifier pairs, both dependency accuracy and dependency coverage of the lexicalized dependency parser were increased by 0.61% when parsing Penn Chinese Treebank. This improvement was regarded as statistically significant (McNemar's test: $p<0.0005$).

Figure 4 shows the dependency trees of an example sentence generated by the baseline model and the proposed approach. In Figure 4(a), the modifier of '以/P' was incorrectly recognized as '奖/NN' and the head of '以/P' was also improperly assigned as '称为/VV' by the baseline model. It was because there did not exist lexicalized dependencies ‘名字/NN→以/P’ and ‘以/P→命名/VV’ in the training data from Penn Chinese Treebank. But in Figure 4(b), ‘名字/NN’ was selected as modifier of ‘以/P’ and ‘命名/VV’ was chosen as head of ‘以/P’ properly by using the head-modifier pairs ‘名字/NN→以/P’ and ‘以/P→命名/VV’ in the proposed approach.

Table 3. *Dpnd.Accu* of different models

Dependency Type	w/o(%)	w/HM(%)
all	82.76	83.37(+0.61)
[N,V]	84.45	85.97(+1.52)
[V,V]	55.42	55.23(-0.19)
[N,N]	84.68	85.19(+0.51)
[V,P]	85.29	87.31(+2.02)
other	86.75	87.00(+0.25)

Table 4. *Dpnd.Cov* of different models

Dependency Type	w/o(%)	w/HM(%)
all	82.76	83.37(+0.61)
[N,V]	84.52	86.04(+0.52)
[V,V]	57.65	57.00(-0.65)
[N,N]	83.10	83.80(+0.70)
[V,P]	86.59	88.11(+1.52)
other	86.78	87.12(+0.34)

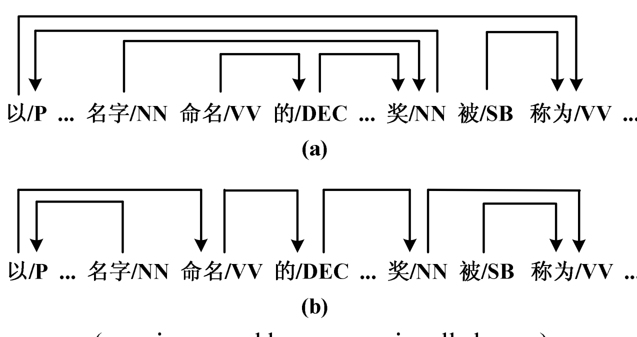

Figure 4. Dependency trees of an example sentence. (a) Dependency tree generated by baseline model; (b) Dependency tree generated by the proposed approach

5.4 Discussion

The experimental results proved that the proposed approach is effective for improving the lexicalized dependency parsing on Penn Chinese Treebank. But there are still some works which should be considered in the future.

ISSN 1736-6305 Vol. 1
http://dspace.utlib.ee/dspace/handle/10062/4476

(1) In the proposed approach, we look sentence as unit and use maximum sentence length as criterion for good parse selection. Then we extract all the head-modifier pairs from the selected good parses as reliable pairs. This method is easy and efficient. But, by using this simple way, we cannot get reliable head-modifier pairs for the dependency type whose accuracy is not good in the deterministic parser.

For example, Table 3 and Table 4 show that both the dependency accuracy and dependency coverage of '[V,V]' type dropped after adding head-modifier pairs into parser. It was because the deterministic parser used for parsing Chinese Gigaword only achieved 65.08% dependency accuracy for '[V,V]' type (see Table 1) in selected sentences.

There are two possible ways to solve this problem. The first way is to change the method for reliable head-modifier pair extraction, such as extracting the head-modifier pairs using the accuracy of different dependency types. The second way is to enhance the good parse selection. For example, Reichart and Rappoport (2007) presented a sample ensemble parse assessment algorithm, which use the level of agreement among several copies of a parser, to predict the quality of a parse. Yates et al. (2006) proposed an algorithm which filters out high quality parses by performing semantic analysis. In our future work, we will try these ways to improve the reliability of extracted head-modifier pairs.

(2) Currently, the probabilities of head-modifier pairs and the probability of dependencies estimated by a treebank are simply multiplied together in the proposed approach (see equation 8). Assigning optimizing weights to different probabilities could be a possible way to enhance the parsing performance. We will consider about this work in the future.

6 Related Work

To our current knowledge, there were few works that apply head-modifier pairs into Chinese lexicalized parsing, except that Wu (2003) proposed an approach for learning the relations between verb and noun to improve parsing. Because of the different testing data set, it is difficult to compare our approach with Wu's work. Roughly speaking, Wu's work focused on the different types of verb-noun relations. But our approach pays attention to dependencies between all kinds of word pairs.

Besides, there have been some works about handling lexical preference by case frame for syntactic analysis or other applications. For example, Kawahara and Kurohashi (2006) integrated automatically constructed Japanese case frames into a Japanese syntactic analyzer and achieved significantly improvement on web sentences. Abekawa and Okumura (2006) introduced the probability of dependency and co-occurrence between verb and its case elements into Japanese dependency parsing. Sasano et al. (2004) used Japanese nominal case frames constructed from a large corpus to help indirect anaphora resolution. While the handling of lexical preference in these approaches were based on case frames, where not only the head-modifier pairs but also the information of case slots are extracted. Compared with these works, our approach only uses the head-modifier pairs to improve lexicalized dependency parsing.

There were also several previous works of dependency parsing on Chinese treebanks. For example, Cheng et al. (2005; 2006) and Hall et al. (2006; 2007) applied shift-reduce deterministic parsing in Penn Chinese Treebank and Sinica Treebank (Chen et al., 2003). Sagae and Tsujii (2007) then generalized the standard deterministic framework to probabilistic parsing by using a best-first search strategy for parsing Sinica Treebank. In these works, lexical preferences were introduced as features for predicting parsing action, which was different from our usage of head-modifier pairs. Besides of them, Wang et al. (2005; 2006) proposed a bottom-up generative parsing model, which was completely lexicalized, to parse Penn Chinese Treebank by decomposing the generation of a parse tree into a sequence of steps. Compared with this work, our proposed approach applied the probabilities of bi-lexical dependencies into probabilistic model to overcome data sparseness problem, rather than introducing word similarity-based smoothing to replace part-of-speech smoothing in Wang et al. (2005; 2006).

7 Conclusion and Future Work

This paper proposed an approach to learn head-modifier pairs, which represent the bi-lexical dependencies, to improve a lexicalized dependency parser for parsing a Chinese treebank. Experimental results show that by using the proposed approach, not only the coverage of bi-lexical dependency was increased, but also the dependency accuracy and dependency coverage of lexicalized dependency parsing were improved significantly. These results proved that the proposed approach can help enlarge the bi-lexical dependency which could not be obtained from treebank data due to data sparseness problem, and the learned bi-lexical dependency can help improve the lexicalized dependency parsing on a Chinese treebank.

While, the proposed approach is only a preliminary work and has much future work to do. The considered future work includes extracting reliable head-modifier pairs by considering about the accuracy of the deterministic parser on different dependency types; enhancing the method for good parse selection; assigning weights to different probabilities in the lexicalized parsing model; and so on.

Acknowledgement

We would like to thank the three anonymous reviewers for their valuable comments.

References

T.Abekawa and M.Okumura. 2006. Japanese Dependency Parsing Using Co-occurrence Information and a Combination of Case Elements. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*. pp. 833-840.

D.Bikel. 2004. Intricacies of Collins' Parsing Model. *Computational Linguistics*. 30(4): 479-511.

H.Cao et al.. 2005. Two-stage Approach to Full Chinese Parsing. *High Technology Letters*. 11(4): 359-363.

E.Charniak. 2000. A Maximum-entropy-inspired Parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*. pp. 132-139.

K.Chen et al.. 2003. Sinica Treebank: Design Criteria, Representational Issues and Implementation. *Abeille(2003), chapter 13*. pp. 231-248.

Y.Cheng, M.Asahara and Y.Matsumoto. 2005. Chinese Deterministic Dependency Analyzer: Examining Effects of Global Features and Root Node Finder. In *Proceedings of IJCNLP 2005*.

Y.Cheng, M.Asahara and Y.Matsumoto. 2006. Multi-lingual Dependency Parsing at NAIST. In *Proceedings of the CoNLL-X*. pp. 191-195.

M.Collins. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. pp. 184-191.

M.Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D Thesis. University of Pennsylvania.

D.Graff et al.. 2005. Chinese Gigaword Second Edition. Linguistic Data Consortium, Philadelphia.

J.Hall, J.Nivre and J.Nilsson. 2006. Discriminative Classifiers for Deterministic Dependency Parsing. In *Proceedings of Coling-ACL 2006*. pp. 316-323.

J.Hall et al. 2007. Single Malt or Blended? A Study in Multilingual Parser Optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. pp. 933-939.

D.Kawahara and S.Kurohashi. 2006. A Fully-lexicalized Probabilistic Model for Japanese Syntactic and Case frame Analysis. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. pp. 176-183.

T.Kudo and Y.Matsumoto. 2002. Japanese Dependency Analysis Using Cascaded Chunking. In *Proceedings of the Conference on Natural Language Learning*. pp. 29-35.

S.Kurohashi and M.Nagao. 1994. A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures. *Computational Linguistics*. 20 (4): 507-534.

T.Nakagawa and K.Uchimoto. 2007. A Hybrid Approach to Word Segmentation and POS Tagging. In *Proceedings of the ACL 2007 Demo and Poster Sessions*. pp. 217-220.

R.Reichart and A.Rappoport. 2007. An Ensemble Method for Selection of High Quality Parses. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. pp. 408-415.

R.Sasano, D.Kawahara, and S.Kurohashi. 2004. Automatic Construction of Nominal Case Frames and its Application to Indirect Anaphora Resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*. pp. 1201-1207.

K.Sagae and J.Tsujii. 2007. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. pp. 1044-1050.

N.Xue, F.Chiou and M.Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 18th International Conference on Computational Linguistics*.

K.Uchimoto et al.. 2000. Dependency Model Using Posterior Context. In *Proceedings of the 6th International Workshop on Parsing Technology*. pp. 321-322.

Q.Wang, D.Schuurmans, and D.Lin. 2005. Strictly Lexical Dependency Parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies* (*IWPT*). pp. 152-159.

Q.Wang et al.. 2006. Improved Larged Margin Dependency Parsing via Local Constraints and Laplacian Regularization. In *Proceedings of the 10th Conference on Computational Natural Language Learning* (*CoNLL-X*). pp. 21-28.

A.Wu. 2003. Learning Verb-Noun Relations to Improve Parsing. In *the 2nd SIGHAN Workshop on Chinese Language Processing*. pp. 119-124.

A.Yates, S.Schoenmackers, and O.Etzioni. 2006. Detecting Parser Errors Using Web-based Semantic Filters. In *Proceedings of EMNLP 2006*. pp. 27-34.

K.Yu, S.Kurohashi, and H.Liu. 2006. Chinese Word Segmentation and Named Entity Recognition by Character Tagging . In *the 5th SIGHAN Workshop on Chinese Language Processing*. pp. 146-149.

K.Yu, S.Kurohashi, and H.Liu. 2007. A Three-step Deterministic Parser for Chinese Dependency Parsing. In *Proceedings of NAACL-HLT 2007*. pp. 201-204.

ISSN 1736-6305 Vol. 1
http://dspace.utlib.ee/dspace/handle/10062/4476