Incorporating Statistical Information of Lexical Dependency into a Rule-Based Parser *

Yoon-Hyung Roh, Ki-Young Lee, and Young-Gil Kim

Natural Language Processing Research Team, Electronics and Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{yhroh, leeky, kimyk}@etri.re.kr

Abstract. This paper presents a method to incorporate statistical information into a rule-based parser to resolve syntactic ambiguities. We extract the statistical information from the Penn Treebank, and apply the information to the rule-based parser. For the extraction of the statistical information the tag conversion is needed because of the disagreement of the tags and the bracketing style. We will show the effect of the tag conversion with experiments. The final result shows about 7% error rate reduction in the dependency evaluation. We will also show how much each type of statistical information affects the parsing performance.

Keywords: rule-based parsing, syntactic ambiguity, statistical information, PCFG, lexical dependency

1 Introduction

While it is easy to develop a rule-based parser and to improve the performance in the early developing stage, it becomes more and more difficult to resolve the conflicts between the rules as the number of rules increases. Because the rule-based parser has the limit in its ability to resolve syntactic ambiguity, syntactic ambiguity is the challenging problem especially for the rule-based parser using CFG as its grammar. One way to solve the problem is lexicalization.

Many recent parsing technologies have taken statistical approaches as we can get more linguistic data such as the Penn Treebank (Collins, 1999; Collins, 2000; Charniak and Johnson, 2005). They also show encouraging performance. But practically the statistical parsing has the efficiency problem and the scalability problem. The scalability problem means the difficulty in incorporating other types of syntactic information such as lexical patterns or semantic patterns. Also it is not easy to tune the parser minutely with respect to each sentence. So, we want to use a PCFG parser as a base parsing system and use statistical information for syntactic ambiguity.

There are many researches about resolving syntactic ambiguity using statistical information. The representative case is PP attachment disambiguation (Stetina and Nagao, 1997; Olteanu and Moldovan, 2005; Foth and Menzel 2006). Most of them simplified the problem into selecting an attachment site between a noun and a verb. However, in the real parsing the situation is more complicated. There can be more attachment sites and the impact of PP attachment on the other part has to be considered. In Foth and Menzel (2006), more comprehensive disambiguation method was presented using Lexical Attraction, a sort of mutual information.

Another related research area is dependency parsing technology (Kudo and Matsumoto, 2000; McDonald *et al.*, 2006). But for the present, we want to add statistical information without significant influence to the current weighting mechanism for the parsed tree selection.

* The work reported in this paper was supported by the IT R&D program of MKE, “Development of Machine Translation Technology for Korean/Chinese/English Spoken Language and Business Documents”.

In the next section we introduce our base parsing system, and in the section 3, we present the way to apply statistical information for syntactic ambiguity. In the section 4, by analyzing the performance variation according to the types of statistical information, we improve the efficiency by applying the statistical information selectively. Then, we conclude this paper with several remarks on future works.

2 Base Parsing System

Our parsing system was developed for the general domain of the web. The parser conducts bottom-up chart parsing using ACFG (Augmented Context Free Grammar) rules, where the rules are constrained on various types of syntactic and semantic contextual condition.

The parsing rules are initially extracted from the Brown Corpus in the Penn Treebank. Not only have the syntactic tags and the bracketing style been modified, but also the rules have been revised and enlarged in the course of extending the target domain of the parser. In the beginning stage, the reason that we didn't adopt a statistical approach is due to the efficiency and the scalability of the parser. Generally, a statistical parser use a huge amount of parameters and the search space is large, so the parsing speed is relatively low and it is not adequate for a real time application. In the preliminary test, the speed of the statistical parsing is more than 7 times lower than that of the rule-based parsing. Also, in the side of scalability, usually a practical parser use various types of additional knowledge for parsing, it is difficult to incorporate other knowledge. Rules are easy to recognize and manage.

Our parser uses lexical patterns and verb subcategorization probability information besides the parsing rules. The rules have many syntactic and semantic features and constraints for prohibiting implausible syntactic structures and prioritizing the rules. Lexical pattern are hand-crafted idiomatic expressions. They include at least one lexical item, and they are applied to sentences comprising the specific word. The verb subcategorization probability information is the probability that a certain verb takes a certain subcategorization type. It was also extracted from the Brown corpus in the Penn Treebank.

Our parser shows over 90% dependency accuracy[1] in the general domain, which is competitive performance in the rule-based parsing. But the parsing performance is standstill and now we need to resolve syntactic ambiguity for additional performance improvement. One way is to use statistical information.

3 Applying Statistical information

When we consider what type of statistical information can be used from the Penn Treebank, the inconsistency between our bracketing style and that of Penn Treebank is an obstacle to using some knowledge like lexicalized rules because it require converting one bracketing style to another. Maybe, the best plausible way to use lexical statistic information is statistical information about lexical dependency. So we mainly consider using lexical dependency information.

3.1 Lexical Dependency Information

There are two different dependency models depending on how the dependency probability is conditioned. One is the bilexical dependency model (Collins, 1996) and the other is the generative model (Collins, 1999). In the bilexical model, given two words w_i, w_h, the probability that w_i is dependent on w_h is expressed as follows.

$$P(D \mid w_i, w_h) \qquad (1)$$

[1] We use our own dependency measure, which will be described later.

In the generative model, given a head w_h, the probability that a dependent w_i is generated is expressed as follows.

$$P(w_i \mid w_h) \tag{2}$$

The state-of-the-art in the statistical parsing is the generative model (Collins, 2000; Charniak and Johnson, 2005). So, first we consider the generative model. The generative probability that the i-th dependent (child) d_i is dependent on a head child h in a chart of a chart parsing is expressed as follows.

$$P(d_i \mid h) \approx P(t(d_i), l(d_i) \mid t(h), l(h), dist(i)) \tag{3}$$

where t(x) represents the tag of x, l(x) represents the lexical root of x, and dist(i) represents the distance feature between the head child and the i-th dependent in the chart.

The distance feature captures how far the dependent is from the head child and it is a function of surface string as in Collins (1999). The problem of using such probability is that the generative probability is too low. This makes it difficult to apply other type of knowledge such as lexical or semantic pattern or to apply the statistical information selectively. The desirable characteristic of the weight of lexical dependency is that the weight has 1 when the head child and its dependent have no preference, has a value greater than 1 when two words have dependency preference, and has a value between 0 and 1 when they have dependency dispreference. For this, we normalize the generative probability by dividing it with the generative probability given only the tag of head child. The dependency weight is expressed as follows.

$$W(d_i \mid h) = \frac{P(t(d_i), l(d_i) \mid t(h), l(h), dist(i))}{P(t(d_i), l(d_i) \mid t(h), dist(i))} \tag{4}$$

Considering that the rule probability of PCFG reflects only the probability about syntactic tag, the weight can be regarded as reflecting the variation by lexicalization. Also, the weight can be expressed another way.

$$W(d_i \mid h) = \frac{P(t(d_i), l(d_i), l(h) \mid t(h), dist(i))}{P(l(h) \mid t(h), dist(i)) * P(t(d_i), l(d_i) \mid t(h), dist(i))} \tag{5}$$

It is the form of the mutual information when P(A) is $P(l(h) \mid t(h), dist(i))$ and P(B) is $P(t(d_i), l(d_i) \mid t(h), dist(i))$.

The above method suffers from the data sparseness problem which the lexical statistical approach usually has. The following back-off method can be used.

$$W(d_i \mid h) = \frac{P(t(d_i) \mid t(h), l(h), dist(i))}{P(t(d_i) \mid t(h), dist(i))} \tag{6}$$

Then the total weight of the rule r applied to a chart is calculated by the following.

$$W(r) = P(r) * \prod_i W(d_i \mid h) \tag{7}$$

Generally, the probability of a parse tree is calculated by multiplying all rules applied to the parsed tree like $P(T) = \prod_i P(r_i)$. Likewise the weight of a parse tree with statistical information is calculated by $W(T) = \prod_i W(r_i)$ and the parse tree with the maximum weight is selected as the final result.

Meanwhile, the dependency probability that the i-th dependent d_i is dependent on a head child h in a chart in the bilexical model is expressed as follows.

$$P(D \mid h, d_i) \approx P(D \mid t(h), l(h), t(d_i), l(d_i), dist(i)) \tag{8}$$

The dependency weight is expressed as follows.

$$W(D \mid h, d_i) = \frac{P(D \mid t(h), l(h), t(d_i), l(d_i), dist(i))}{P(D \mid t(h), t(d_i), dist(i))} \tag{9}$$

When we cannot find the probability $P(D \mid t(h), l(h), t(d_i), l(d_i), dist(i))$, it can also be backed off as follows.

$$W(D \mid h, d_i) = \frac{P(D \mid t(h), t(d_i), l(d_i), dist(i))}{P(D \mid t(h), t(d_i), dist(i))} \tag{10}$$

$$W(D \mid h, d_i) = \frac{P(D \mid t(h), l(h), t(d_i), dist(i))}{P(D \mid t(h), t(d_i), dist(i))} \tag{11}$$

Actually we conducted preliminary test about the bilexical model and it showed lower performance than the generative model. Besides, it generates too many parameters because the sample space is all combination of two words in a sentence. So we will consider only the generative model henceforth.

3.2 Extracting Lexical Dependency Information

We use the Penn Treebank as the linguistic data source. When we extract the dependency data from the Penn Treebank, there are several points to consider.

The first is that the Penn Treebank uses some coarse tags. For example, the part of speech (POS) tag "IN" includes both prepositions such as "in" and conjunctions such as "while". Also the Treebank does not distinguish the "TO" of a preposition and the "TO" of a to-infinitive. Moreover, SBAR represents all types of clauses including noun clauses such as that-clause, adverbial clauses such as if-clause, and relative clauses such as which-clause. For this problem, we do not use a POS tag but the syntactic tag of the parent of the pre-terminal in the syntactic tree.

The second is the problem by the difference of syntactic tags and bracketing style. Our parser basically uses the Penn Treebank tags, but we modified syntactic and POS tags and modified the bracketing style from the Penn Treebank. For example our parser distinguishes adverbial clauses (SBARV) such as if-clause from that-clause (SBAR). So, the porting of tags and structures is needed .

Lastly, using syntactic tag eliminates some important information. In case of a verb, the syntactic tag “VP”(Verb Phrase) misses the form information of the verb such as an ing-form or an infinitive form. So we distinguish them by using different tags such as VPG(present participle VP), VPB(infinitive VP), VPN(past participle VP), etc, only in the case that the verb is used as a dependent. In the case that the verb is used as a head, we do not distinguish them.

The overall procedure is as follows:

- All pre-terminals in the parse tree are recognized. For all the pre-terminals, conduct the following.
  For example, from the below parse tree, “He/PRP accused/VBD Dow/NNP Jones/NNP of/IN using/VBG unfair/JJ means/NNS..” is extracted.
  ```
  (SS (S
     (NP-SBJ (PRP He) )
     (VP (VBD accused)
       (NP-1 (NNP Dow) (NNP Jones) )
       (PP-CLR (IN of) (`` ``)
         (S-NOM
           (NP-SBJ-2 (-NONE- *-1) )
           (VP (VBG using)
             (NP (JJ unfair) (NNS means) )
  …
     (. .) ('' '') ))
  ```
- word/tag normalization: words are stemmed, the words tagged with “CD, NNP” are replaced by their tags, and the tags “VBZ VBD” are replaced by “VBP” for coverage, etc. (he/PRP accuse/VBP NNP/NNP NNP/NNP of/IN use/VBG unfair/JJ mean/NNS..)
- Finding its head in the pre-terminals using tree structures.
  (accuse/VBP! NNP/NNP of/IN)
- Tag conversion: As described above, some POS tags or syntactic tags are converted.
  ```
  (VPG (VBG using)
            (NP (JJ unfair) (NNS means) )
   (PP-DIR (IN to)
            (NP (NN single-A-3) ))
  ```
- Count all the events of word pairs with the distance feature.
  From (accuse/VBP! NNP/NNP of/IN), the followings are generated.
  000 accuse/VBP! NNP/NNP
  100 accuse/VBP! of/IN
  100 accuse/VBP
- Calculate all the lexical dependency weights according to the formula (4).

$$W(of\,/\,IN \mid accuse\,/\,VBP,100)=\frac{P(IN,of \mid VBP,accuse,100)}{P(IN,of \mid VBP,100)}$$

There are several tag usage strategies:

- Tag1: Using the POS tags of the terminal nodes.
  From the parse tree (VP (VBG using) …), “use/VBG”
- Tag2: Using the parent tags of the terminal nodes. (“use/VP”)
- Tag3: Tag2 with reflecting the above considerations. (“use/VPG”)

For the experiment, we use the standard data division (Collins, 1999). The lexical data was extracted from the section 02-21 of the WSJ corpus. And the section 23 was reserved for the evaluation and the section 00 is used as a development set. The total number of extracted dependency information is 60,550. Some extracted samples are shown below:

39.9245 89 000 account/VP! for/PP
1.6473 3 000 accountable/ADJP! PP
19.4185 3 000 accountable/ADJP! for/PP
1.7168 4 000 accrue/VP! PP
0.8807 4 000 accusation/NP! PP
1.2503 3 000 accusation/NP! of/PP
110.5855 18 000 accuse/VP! of/PP
3.4336 3 000 accustom/VP! PP
26.1406 3 000 accustom/VP! to/PP

In the above example, the word with "!" is a head and the first field is the dependency weight. For example, "110.5855 18 000 accuse/VP! of/PP" means that the dependency weight that the "of/PP" comes immediately after the head "accuse/VP" is 110.5855.

3.3 Applying Dependency Weight

When an inactive chart is generated in the chart parsing, all the dependency weights between the head child and other children calculated and multiplied to the total chart weight. But we exclude some children which have little dependency ambiguity like "the".

3.4 Evaluation

The common method to evaluate the parsing performance is the way by matching bracketing. But that method is affected heavily by the tags and bracketing style. Moreover, our parser uses the lexical patterns, which does not follow usual recognition unit of syntactic constituent like "IN -> in reference with", "VB -> provide NP with". So we use the dependency accuracy between words.

All words in a sentence have their own headword except the headword of the whole sentence. Therefore, the performance of dependency accuracy is measured by obtaining the headwords and matching them. The usual dependency accuracy includes the match of the relation between the head and its dependents (Lin, 1998). But we do not consider the tags or any relation because of the disagreement of tags and bracketing style. Our method only discerns whether the dependent is an argument or not. This makes it possible to distinguish whether to-infinitive is used as an argument of a verb such as "want" or not.

Table 1 shows the parsing performance by dependency accuracy.

Table 1: The dependency accuracy of parsing results.

	Labeled Precision	Dependency Accuracy	Error Rate Reduction
Collins Model2 (Collins, 1999)	88.3%	91.00%	
Base parsing system		91.27%	
Tag1		91.79%	5.95%
Tag2		91.72%	5.15%
Tag3		91.89%	7.10%
Tag3 with back-off		91.80%	6.07%

For comparison, Table 2 shows all the parsing performance with respect to the tag usage strategy. The Tag3 method shows about 7% error rate reduction. Contrary to our expectation the Tag3 with back-off shows performance degradation. It seems mainly due to the coarse granularity of syntactic tags for the back-off model.

4 Analysis on the effect according to Lexical Dependency Information Type

There are two reasons of analyzing the performance variation of each dependency type. In the parsing, the application of the statistical information causes the efficiency problem. So we want to know what type of statistical information affects the parsing performance most little because not all types of statistical information seem to contribute to the dependency performance. For example the direct object of a verb has little ambiguity of dependency. And we do not apply such type of statistical information.

Also, we want to know what type of dependency information we need to build. Building the linguistic knowledge manually is an expensive task. So we want to know what type of dependency information is most effective to the parsing performance. The dependency type is obtained by categorizing the dependency by the tag pair of the dependency information. We can test the selectional preference strength about each lexical dependency (Brockmann and Lapata, 2003).

$$S(v) = \sum_{c \in C} P(c|v,r) \log \frac{P(c|v,r)}{P(c)}$$

where S(v) is a selectional preference strength of a verb v, P(c) is the overall distribution of classes which the verb takes as the relation r. P(c|v,r) is conditional probability. We can get the selectional preference strength of each dependency type by replacing v, r with a head, c with the dependents of the head.

But for now, we want to know the direct effect to the parsing performance because though direct objects have high selectional preference, they do not seem to contribute to the performance enhancement. Table 2 shows the performance variation according to dependency type.

Table 2: Performance variation according to dependency type

Lexical dependency type	Dependency Accuracy	Lexical dependency type	Dependency Accuracy
NP! PP	91.54%	SINV VP!	91.29%
VP! PP	91.39%	SS! VPN	91.29%
NP! VPF	91.39%	PP VP!	91.29%
ADJP! PP	91.33%	SBAR! VP	91.29%
VP! VPF	91.32%	NP! WHNP	91.29%
VP! SBAR	91.30%	VP! S	91.29%
NP! VPN	91.30%	VP! VPG	91.29%
NP NP!	91.29%	WHNP! VP	91.29%
PP! NP	91.29%	VP! SBARV	91.29%
NP VP!	91.29%	SS! VPB	91.29%
VP VP!	91.29%	VPB VP!	91.29%
SS! VP	91.29%	ADJP NP!	91.29%
S VP!	91.29%	VP! VPN	91.29%
NP! NP	91.29%	VP! PRT	91.29%
VP! VP	91.29%	PP! PP	91.29%
ADVP VP!	91.29%	PP! VPG	91.29%
VP! ADJP	91.29%	VP! NP	91.28%
ADJP ADJP!	91.29%	VP! VPB	91.28%
VP! ADVP	91.29%	Base	91.27%

As the Table 2 shows, the statistical information types which contribute to parsing performance is mainly by the attachment of PP, to-infinitive(VPF), SBAR, past participle VP(VPN), etc.

5 Conclusion

This paper presented a method to incorporate statistical information into a rule-based parser to resolve syntactic ambiguity. We employ a PCFG parser as a base system and use additional lexical knowledge for the syntactic disambiguation. We extracted the statistical information from Penn Treebank, and applied the information to the rule-based parser. The result shows about 7% error reduction in the dependency evaluation.

We also conducted some analysis about how much each type of statistical information affects the parsing performance, thus applying the statistical information selectively for efficiency. This analysis result can be used for building additional information for syntactic disambiguation.

For the future works, we need to analyze the sentences whose parsing accuracy is lower than statistical parsing result and what information need to be reflected. And we should analyze why the backed-off method deteriorates the performance. Lastly, we plan to add the verb subcategorization type to the condition of the generative model to cope with data sparseness.

References

Brockmann, C. and M. Lapata. 2003. Evaluating and Combining Approaches to Selectional Preference Acquisition. *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*.

Charniak, E. and M. Johnson. 2005. Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking. *Proceedings of 43rd Meeting of Association for Computational Linguistics*, pp. 173-180.

Collins, M. 1996. A New Statistical Parser based on Bigram Lexical Dependencies. *Proceedings of ACL'96*, pp. 184–191.

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

Collins, M. 2000. Discriminative Reranking for Natural Language Parsing. *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 175–182.

Foth, K. and W. Menzel. 2006. The benefit of stochastic PP attachment to a rule-based parser. *Proceedings of the COLING/ACL*, pp. 223–230.

Kudo, T. and Y. Matsumoto. 2000. Japanese Dependency Structure Analysis Based on Support Vector Machines. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, pp. 18–25

Lin, D. 1998. A Dependency-based Method for Evaluating Broad-coverage Parsers. *Natural Language Engineering*, 4(2), 97-114.

McDonald, R., K. Lerman and F. Pereira. 2006. Multilingual Dependency Analysis with a Two Stage Discriminative Parser. *Proceedings of the Tenth Conference on Computational Natural Language Learning*.

Olteanu, M. and D. Moldovan, 2005. PP-attachment Disambiguation Using Large Context. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 273-280

Stetina, J. and M. Nagao. 1997. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. *Proceedings of the Fifth Workshop on Very Large Corpora*, pp. 66–80.