Improving PP Attachment Disambiguation in a Rule-based Parser ∗

Yoon-Hyung Roh[a], Ki-Young Lee[a], and Young-Gil Kim[a]

[a]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 deals with how to enhance the performance of a rule-based parser using statistical Information. PP (Prepositional Phrase) attachment ambiguity is one of the main ambiguities found in parsing. We therefore conducted some experiments on extracting statistical information for PP attachment from a corpus, and on applying such information to a rule-based parser. Two types of information are used: supervised learning data and unsupervised learning data. In this paper, we show how we apply these types of information and to what degree they contribute to the PP attachment as well as to the overall parsing performance. The final results show a 5.42% performance improvement in PP attachment, with an 8.7% error reduction ratio in the overall parsing performance.

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

1 Introduction

For a rule-based parser using CFG as its grammar, syntactic ambiguity is a challenging problem as it becomes more difficult to resolve conflicts between rules as the number of rules increases. Certain methods to solve this problem have been suggested, and have generally introduced statistical approaches (Sagae and Lavie, 2003; Foth and Menzel, 2006).

Many recent parsing technologies have also taken statistical approaches, as we can get more linguistic data, such as the Penn Treebank (Collins, 1999; Collins, 2000; Charniak, 2000). These technologies have shown encouraging performances. But practically speaking, statistical parsing has efficiency and scalability problems. As a statistical parser generally uses a significant number of parameters, and the search space is large, the parsing speed is relatively low and is not adequate for a real-time application. In our previous research, we conducted some experiments on applying statistical information to a rule-based parser (Roh, 2009). The parsing performance was affected mainly in the attachment performance and by the attachment information. Therefore, we do not need to apply full statistical information.

The scalability problem indicates the difficulty in incorporating other types of knowledge for use in parsing. Usually, a practical parser uses various types of additional knowledge for parsing, such as lexical patterns or semantic patterns. However, it is not easy to incorporate other knowledge into a parsing system based on probability. It is also difficult to tune the parser minutely with respect to each sentence. Thus, we want to use a rule-based parser as a base parsing system, and selective statistical information for ambiguity resolution.

Our parsing system conducts bottom-up chart parsing using ACFG (Augmented Context Free Grammar) rules, which have many syntactic and semantic features and constraints for

∗ 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”.

prohibiting implausible syntactic structures and prioritizing the rules. These parsing rules are initially extracted from the Brown Corpus in the Penn Treebank. Not only have the syntactic tags and bracketing style been modified, but the rules have also been revised and enlarged in the course of extending the target domain of the parser. Our parser shows a competitive performance in rule-based parsing. However, the parsing performance remains at a standstill, and we now need to resolve syntactic ambiguity for additional improvements in performance.

In the next section, we briefly introduce a PP attachment task, and in sections 3 and 4, we present ways to apply supervised learning data and unsupervised learning data for a PP attachment, respectively. In section 5, we show a method for applying two types of data. Finally, we conclude this paper with several remarks on our future work.

2 PP attachment Disambiguation

PP attachment disambiguation is a task used to determine which site a PP is attached to while parsing. There have been many researches regarding PP attachment disambiguation (Ratnaparkhi, 1998; Stetina and Nagao, 1997; Olteanuand and Moldovan, 2005; Foth and Menzel, 2006). The main issues are mainly concerned with data extraction, learning methods, disambiguation models, and the treatment of the data sparseness problem.

Most of them simplify the problem into selecting an attachment site between a noun and a verb. However, in real parsing, the situation is more complicated. There can be more attachment sites, and the impact of a PP attachment on the other part has to be considered.

A few researchers have dealt with multiple PP attachments (Merlo, 1997; Bharati, 2005). Foth and Menzel (2006) presented a more comprehensive disambiguation method using Lexical Attraction, which is a type of mutual information.

We solve this problem by simply giving an attachment weight to the existing chart weight. We focus on how we can calculate the attachment weight when a PP is attached to some other constituent.

3 Supervised Learning

In conducting supervised learning, we consider a PP attachment task as a process of full syntactic analysis, and we use the statistical model typically used in statistical parsing.

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

$$P(D \mid w_i, w_h), \tag{1}$$

where D represents a dependency relation between w_i and w_j.

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

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

3.1 Generative Model

The state of the art model in statistical parsing is the generative model (Collins, 2000; Charniak, 2000). Therefore, we first consider this model in our analysis. The generative probability that the i-th dependent (child) d_i is dependent on head child h in a chart generated in 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 i-th dependent in the chart.

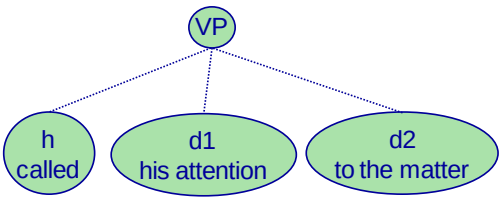

Figure 1: An example chart

Figure 1 shows a snapshot of the parsing process of the sentence, "President Bush called his attention to the matter." In the chart, the generative probability is as follows.

$$P(d_2 \mid h) = P(to, TO \mid call, VBD, 100) \quad (4)$$

The distance feature "100" captures how far the dependent is from the head child, and is a function of the 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 types of knowledge such as lexical or semantic patterns, or to apply statistical information selectively. The desirable characteristic of the lexical dependency weight is that the weight is 1 when the head child and its dependent have no preference, is a value greater than 1 when two words have a dependency preference, and is a value between 0 and 1 when they have a dependency dispreference. For this, we normalize the generative probability by dividing it with the generative probability given only the tag of the head child. The dependency weight is expressed as follows.

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

Considering that the rule probability of PCFG reflects only the probability about a syntactic tag, the weight can be regarded as reflecting the variation through lexicalization.

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

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

The total weight of rule *r* applied to a chart is then calculated by the following.

$$W(r) = P(r) * \prod_i W(d_i \mid h) \quad (7)$$

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

3.2 Bilexical Model

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

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

In the chart in figure 1, the dependency probability is as follows.

$$P(D \mid call, to) = P(D \mid call, VBP, to, TO, 100)$$

As in the case of the generative model, the probability has to be applied in the form of a weight. The dependency weight is expressed as follows.

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

This shows the ratio of the probability conditioned on particular lexical items to the probability conditioned on a more generalized tag.

In previous research, we conducted experiments on these two models and obtained better results from the generative model (Roh, 2009). However, in analyzing the errors in detail, we found that the dependency weight of the bilexical model has some problems.

First, the ratio is not adequate in this situation. For example, for *P(D|h,p)* = 1.0 and *P(D|t(h1),t(p))* = 0.5, the weight is 2.0, while for *P(D|h2,p)* = 0.9 and *P(D|t(h2), t(p))* = 0.3, the weight is 3.0. Here, *p* and *h2* have greater weight, but it is obvious that *p* and *h1* must have more weight based on the definition of bilexical dependency probability. In other words, the dependency weight should be purely dependent on the bilexical dependency probability, not on the ratio.

Second, *p* and *h1* need to have much greater weight as *P(D|h,p)* approaches 1. Thus, the weight should not be linearly proportional to *P(D|h,p)*. With this in mind, we can revise the expression as follows.

$$W(D \mid h, d_i) = (P(D|h, d_i) + 0.5)^n \quad (10)$$

In the above expression, when $P(D \mid h, d_i)$ = 0.5, the weight becomes 1.0, which has no effect as a weight. This is reasonable since we are unable to give either preference or dispreference if the probability that *h* and *p* have a dependency relation is one half. The value *P(D|h,p)+0.5* then has a range of 0.5 to 1.5, and the weight decreases or increases as it distances itself from 1.0. The value *n* was determined experimentally, and we use 7.0 as its value.

3.3 Extracting Lexical Dependency Information

We use the Penn Treebank as the linguistic data source (Marcus, 1993). When extracting the dependency data from the Penn Treebank, there are certain points that we need to consider.

In the Penn Treebank, some coarse tags are used. 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 from the "TO" of a to-infinitive. Moreover, SBAR represents all types of clauses including noun clauses such as that-clauses, adverbial clauses such as if-clauses, and relative clauses such as which-clauses. In addition, while our parser basically uses the Penn Treebank tags, we had to modify the syntactic and POS tags, which resulted in some mismatched tags. For example our parser distinguishes adverbial clauses (SBARV) such as if-clauses from that-clauses (SBAR).

Lastly, using a syntactic tag eliminates some important information. In the case of a verb, the syntactic tag "VP" (Verb Phrase) misses the form information of the verb, for example, whether it is in the gerund form or infinitive form. Thus, we distinguish between them by using different tags such as VPG (present participle VP), VPB (infinitive VP), VPN (past participle VP), etc, only when the verb is used as a dependent. When the verb is used as a head, we do not distinguish them.

For this problem, we do not use a POS tag but rather the syntactic tag of the parent of the pre-terminal in the syntactic tree, and we conduct a tag conversion.

The overall procedure is as follows:

- Find its heads in the pre-terminals using a tree structure. The heads are marked by '!'.

```
(SS (S!
  (NP (NNP President) (NNP Bush) )
  (VP! (VBD! called)
```

(NP (PRP$ his) (NN! attention))
(PP (TO! to)
(NP (DT the) (NN! matter))))
(. .)))

For example, from the parse tree below, "President/NNP Bush/NNP called/VBD his/PRP$ attention/NN to/TO the/DT matter/NN" is extracted.

- Conduct word/tag normalization, where the words are stemmed, and those tagged with "CD, NNP" are replaced by their tags, while the tags are replaced by their parent tags:
  (Bush/NNP called/VBD!) => (NNP/NNP called/VBD!)
- Count all the events of word pairs with the distance feature.
  From (call/VP! attention/NP to/PP), the following are generated.
  000 call/VP! attention/NP
  100 call/VP! to/PP
  100 call/VP
- Calculate all the lexical dependency weights according to the formulas used for the dependency weight.

3.4 Experiments

For the experiment, we use the standard data division (Collins, 1999). The lexical data was extracted from section 02-21 of the WSJ corpus. In addition, section 23 was reserved for the evaluation, and section 00 was used as a development set.

The common method used to evaluate the CFG style parsing performance is through matching bracketing. However, this method is heavily affected by the tags and bracketing style. Therefore, we use the dependency accuracy between words. Except for the headword of a sentence as a whole, all words in the sentence have their own headword. Therefore, the dependency accuracy is measured by obtaining and matching the headwords. The usual dependency accuracy includes the match of the relation between the head and its dependents (Lin, 1998). However, we do not consider the tags or any relation due to the tag disagreement and bracketing style. The dependency performance is measured by obtaining how many words are attached to its correct headword. Table 1 shows the results.

Table 1: Attachment accuracy of supervised learning

	PP attachment Accuracy	Dependency Accuracy
Base parsing system	79.23%	91.39%
generative model	83.04%	91.92%
bilexical model	83.44%	91.97%

Table 1 shows that the bilexical model is slightly better than the generative model. This is considered reasonable since the PP attachment problem is to find the attachment site when a sentence is given, and is better described by the bilexical model. The baseline for the experiments in this paper is our base parsing system, as our main concern is not PP attachment itself, but how we can improve the parsing performance using a statistical PP attachment disambiguation method.

4 Unsupervised Learning

4.1 Data Extraction

In an unsupervised learning method for PP attachment, the learning data is usually extracted heuristically (Ratnaparkhi, 1998). We followed a similar method. For the unsupervised learning

data, the raw corpus is tagged and chunked, and all chunked NPs are replaced with their head words. Next, 4-tuples (*v*, *n*, *p*, *n2*) are extracted, where *v* is the closest verb occurring on the left of preposition *p*, and there is no conjunction except for a coordinate conjunction between v and *p*. In addition, *n* is the closest noun occurring on the left of preposition *p* with no verb between *n* and *p*. Finally, *n2* is the object noun of *p*. If there is not either *v* or *n*, it is denoted as *NULL*.

For example, the sentence "A man in the park saw a girl with a telescope" is tagged and chunked as follows:

A/DT man/NN in/IN the/DT park/NN saw/VBD a/DT girl/NN with/IN a/DT telescope/NN ./.
=>
man/NN in/IN park/NN saw/VBD girl/NN with/IN telescope/NN ./.

4-tuples: (*NULL*, man, in, park), (see, girl, with, telescope)

The number of triple (*v*/*n*, *p*, *n2*) is then counted as follows according to the ambiguity.

1) If a 4-tupe has the form of (*v*, *NULL*, *p*, *n2*), then add 1 to the number of triple (*v*, *p*, *n2*).
2) If a 4-tupe has the form of (*NULL*, *n*, *p*, *n2*), then add 1 to the number of triple (*n*, *p*, *n2*).
3) If a 4-tupe has the form of (*v*, *n*, *p*, *n2*), then several counting methods are considered.

Counting method 1: Add 0.5 to the number of triple (*v*, *p*, *n2*) and 0.5 to the number of triple (*n*, *p*, *n2*).

Counting method 2: Add *P(D|VERB, PREP)* to the number of triple (*v*, *p*, *n2*) and *P(D|NOUN, PREP)* to the number of triple (*n*, *p*, *n2*).

Counting method 3: Add *P(D|VERB, p)* to the number of triple (*v*, *p*, *n2*) and *P(D|NOUN, p)* to the number of triple (*n*, *p*, *n2*).

Counting method 4: Add *P(D|v, p)* to the number of triple (*v*, *p*, *n2*) and *P(D|n, p)* to the number of triple (*n*, *p*, *n2*).

In the above example sentence, the numbers of triples are as follows.

Counting method 1:

(man, in, park):1, (see, with, telescope):0.5, (girl, with, telescope): 0.5

Counting 2:

(man, in, park):1, (see, with, telescope):P(D|VERB, PREP) = 0.395, (girl, with, telescope): P(D|NOUN, PREP) = 0.691

Counting 3:

(man, in, park):1, (see, with, telescope): P(D|VERB, with) = 0.527, (girl, with, telescope): P(D|NOUN, with) = 0.632

For the linguistic source, the BLLIP WSJ corpus was used, which is composed of about 1.8 million automatically tree-tagged sentences. The pos-tagged results were extracted, the results of which were used to extract the triples. The number of triples extracted was 1,466,472.

4.2 Model

For the attachment weight, we used the generative model and its ratio to a more general case conditioned on tags as follows.

$$W(p,n2 \mid w) = \lambda \frac{P(p,n2 \mid w)}{P(p,n2 \mid t(w))} + (1-\lambda)\frac{P(p \mid w)}{P(p \mid t(w))}, \tag{11}$$

where *w* is a verb or a noun, *t(w)* is the tag of *w*, and λ is an interpolation constant for data sparseness. The method used to apply the weight is similar to that of the supervised learning.

$$W(r) = P(r) * \prod_i \alpha W(p,n2 \mid w), \tag{12}$$

where α is a proportional constant used for scaling the PP attachment weight in order to adjust the weight according to the parsing weighting mechanism, and is determined experimentally.

4.3 Experiments

As in the experiments on supervised learning, the evaluation was conducted using section 23 of the WSJ corpus. Table 2 shows the results.

Table 2: Attachment accuracy of unsupervised learning according to the counting methods of ambiguous data

	PP attachment Accuracy	Dependency Accuracy
Base parsing system	79.23%	91.39%
Counting method 1	82.33%	91.70%
Counting method 2	81.26%	91.65%
Counting method 3	81.71%	91.69%
Counting method 4	81.68%	91.70%
(v, p, n2): (n, p, n2) = 0.7:0.3	82.95%	91.78%

One puzzling result is that counting method 1 is better than the other cases where supervised learning data are applied. We assumed that a more informed case would have better results than the case of simply giving one half. Therefore, we conducted additional experiments, changing (*v*, *p*, *n2*) using the constant values of 0.6, 0.7, and 0.8. As a result, 0.7 gives the best results, and was thus selected as the final value. However, further analysis regarding these results is needed.

5 Combination

We considered two methods in combining supervised and unsupervised learning data.

1) We applied supervised learning data when supervised learning data was available for the attachment weight, and we applied unsupervised learning data only when there was no supervised learning data available (super > unsuper).
2) We applied the attachment weight using both supervised and unsupervised learning data at the same time (super + unsuper).

Table 3 shows the results.

Table 3: Attachment accuracy of combining supervised and unsupervised learning methods.

	PP attachment Accuracy	Dependency Accuracy
Base parsing system	79.23%	91.39%
supervised	83.44%	91.97%
unsupervised	82.95%	91.78%
super > unsuper	84.06%	92.09%
super + unsuper	84.65%	92.14%

Volk (2002) took the "super > unsuper" approach. It seems that these cases are different as they deal with PP attachment as a classification problem, while we deal with it as a weighting problem. Finally, using statistical information, the PP attachment performance was enhanced by 5.42% with an 8.7% error reduction ratio in the overall parsing performance.

6 Conclusion

This paper presented a method to enhance the PP attachment performance in a rule-based parser. We conducted a comparative experiment between the bilexical and generative models for

supervised learning. We also applied the unsupervised learning method of PP attachment to the rule-based parser. The results show a 3.73% improvement in performance.

We also conducted simple experiments on combining two types of learning methods, the final results of which show a 5.42% performance improvement in PP attachment, with an 8.7% error reduction ratio in the overall parsing performance.

For future work, we obviously need to conduct more analysis and experiments on adding counts to ambiguous 4-tuples by applying the supervised learning method. In addition, we need a more elaborate method to combine the supervised and unsupervised learning methods. Finally, we plan to devise an efficient way to build a tagged corpus for PP attachment.

References

Bharati, A., U. Rohini, P. Vishnu, S. Bendre, and R. Sangal. 2005. A hybrid approach to single and multiple PP attachment using WordNet. *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP) 2005*, pp. 211–222.

Charniak, E. 2000. A maximum-entropy-inspired parser. *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 132-139.

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

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.

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.

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

Marcus, M. P., B. Santorini, and M. A. Marcinkiewics, 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19.

Merlo, P., M. W. Crocker, and C. Berthouzoz. 1997. Attaching multiple prepositional phrases: Generalized backed-off estimation. *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, pp. 149–155.

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.

Ratnaparkhi, A. 1998. Statistical Models for Unsupervised Prepositional Phrase Attachment. *Proceedings of the 36th conference on Association for Computational Linguistics*, pp. 1079–1085.

Roh, Y.-H., K.-Y. Lee, and Y.-G. Kim, 2009. Incorporating Statistical Information of Lexical Dependency into a Rule-Based Parser. *Proceedings of the 23th Pacific Asia Conference on Language, Information and Computation*, pp. 493–500.

Sagae, K. and A. Lavie, 2003. Combining Rule-based and Data-driven Techniques for Grammatical Relation Extraction in Spoken Language. *Proceedings of the Eighth International Workshop in Parsing Technologies*.

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.

Volk, M. 2002. Combining unsupervised and supervised methods for PP attachment disambiguation. *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 1–7.