NEU Systems in SIGHAN Bakeoff 2012

Ji Ma, Longfei Bai, Ao Zhang, Zhuo Liu and Jingbo Zhu

Natural Language Processing Laboratory,
Northeastern University, Shenyang, Liaoning, China.
majineu@outlook.com
{longfeibai|liuzhuo|zhangao}@ics.neu.edu.cn
zhujingbo@mail.neu.edu.cn

Abstract

This paper describes the methods used for the parsing the Sinica Treebank for the bakeoff task of SigHan 2012. Based on the statistics of the training data and the experimental results, we show that the major difficulties in parsing the Sinica Treebank comes from both the data sparse problem caused by the fine-grained annotation and the tagging ambiguity.

1 Introduction

Parsing has been a major interest of research in the NLP community. For the last two decades, statistical approaches to parsing achieved great success and parsing performance has been significantly improved. One of the most important factors for developing accurate and robust statistical parsers for one language is the availability of large scale annotated Treebank in that language. The availability of the Sinica Treebank provides such an opportunity for developing statistical parsers for traditional Chinese.

In this paper, we analyze the difficulties in parsing the Sinica Treebank. By comparing the statistics between the Sinica Treebank and CTB we found that the fine-grained annotation schema adopted by the Sinica Treebank lead to more severe data sparse problem. By inspecting the parsing results, we also found that a great portion of parsing errors is caused by tagging errors. In particular, word classes such as Ng and Ncd are quite similar in their meaning. However, the two tags yield quite different syntactic structures.

2 Parsing Models

The probabilistic context free grammar is the basis for a great portion of parsing approaches developed in the last two decades. However, the vanilla probabilistic context free grammar achieves poor performance. This is due to its strong independence assumptions which lead to decisions made by the PCFG model extremely local thus lacks of discriminative power. In terms of weakening the independence assumption of the PCFG model, the approaches adopted by modern state-of-the-art parsers can be roughly divided into two categories.

Head driven methods or lexicalized methods (Collins, 1999; Charniak 2000) augment the PCFG model with bi-lexical dependencies, subcategorization frames and other information such direction and surface distances. With those information, the lexicalized parsers can make more informed decisions and parsing performance significantly improved over the vanilla PCFG model. However, the head driven methods may not suitable for the current task for two reasons. (1) to acquire the bi-lexical dependencies, a set of manually collected head finding rules are needed. To our knowledge, there is not such set of rules for the Sinica Treebank. (2) some of the information utilized by the head-driven model are specially designed for the Penn Treebank annotation scheme and when shifted to other annotation schemes, parsing performance dramatically decreases (Guldea, 2001).

Rather than using the bi-lexical dependencies, unlexicalized methods (Klein and Manning 2003; Matsuzaki et al., 2005; Petrov et al., 2006) augment the non-terminals of the PCFG model with latent annotations, PCFGLA hereafter. Those latent annotations are aimed to capture different behavioral preferences of the same non-terminal or production rule in different local context. For example, verb phrases are further split into several subcategories that capture the behavioral preference of infinitive VPs, passive VPs and intransitive VPs. With those latent annotations, parsing performance is greatly improved. Among the unlexicalized methods listed above, the one proposed by Petrov et al., (2006) can learn the latent annotations in a fully automatic manner.

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 206–210,
Tianjin, China, 20-21 DEC. 2012

Compare with the lexicalized methods, their approach does not rely on any head finding rules or corpus specific heuristics. Moreover, their approach consistently outperforms the lexicalized methods across corpus and languages (Petrov and Klein, 2007).

Thus, we choose the PCFG-LA proposed by Petrov et al., (2006) to be our model for the traditional Chinese parsing task.

2.1 A Brief Review of PCFG-LA

In this subsection we briefly review the method of Petrov et al., (2006). Petrov et al., (2006) learns a sequence of PCFG-LA models (G_0, G_1, ... ,G_6) in an iterative manner. The initial grammar G_0 is the one directly read off the Treebank with right binarization. In the *i-th* iteration, their method performs the following three subprocedures:

Split: Each non-terminal are split into two new symbols. For example, suppose T is the parse tree of sentence S in the training corpus. F is a non-terminal in T and F generates span (r, t). L and R are also non-terminals in T. L and R generates span (r, s) span (s, t), respectively. After splitting, F is split into F1 and F2, L is split into L1 and L2, R is split into R1 and R2. The parameters are estimated using a variant of the EM algorithm. Specifically, the inside-outside probabilities can be computed as:

$$P_{in}(F_x, r, t) = \sum_{m,n} \beta(F_x \to L_m R_n) * P_{in}(L_m, r, s) * P_{in}(R_n, s, t) \quad (1)$$

$$P_{out}(L_m, r, s) = \sum_{x,n} \beta(F_x \to L_m R_n) * P_{out}(F_x, r, t) * P_{in}(R_n, s, t) \quad (2)$$

$$P_{out}(R_n, s, t) = \sum_{m,x} \beta(F_x \to L_m R_n) * P_{out}(F_x, r, t) * P_{in}(L_m, r, s) \quad (3)$$

Where β denotes the rule probabilities and the indexes m, n and x are all ranging from 1 to 2. In the E step, the partial count of the rule $P_x \to L_m R_n$ in T can be computed as

$$C(F_x, r, s, t \to L_m R_n) \propto P_{out}(F_x, r, t) * P_{in}(L_m, r, s) * P_{in}(R_n, s, t) \quad (4)$$

In the M step, the partial counts are used to re-estimate rule probabilities:

$$\beta(F_x \to L_m R_n) = \frac{C(F_x \to L_m R_n)}{\sum_{m,n} C(F_x \to L_m R_n)} \quad (5)$$

Merge: To control the grammar size, and also to prevent overfitting, in the merging stage, only the most important splits are reserved and all the others are merged back to the annotation before splitting. The importance of split each non-terminal is measured according to the loss of likelihood after merging it. Large loss denotes more important split therefore should be reserved. Petrov et al., (2006) adopted an efficient way to approximate the likelihood loss after merging each pair of new annotation.

Suppose T is the parse tree of sentence S in the training corpus. F is a non-terminal in T and F generates span (r, t). Suppose that in the i-th iteration, F is split into several new symbols F1, F2,.., Fk. The likelihood of the training data, the sentence-tree pair (S, T), can be computed using the inside-outside probability as

$$LL(S, T) = \sum_x P_{in}(F_x, r, t) * P_{out}(F_x, r, t) \quad (6)$$

Consider that we are going to merge F1 and F2 into F0, then the inside and outside probability are computed as:

$$P_{in}(F_0, r, t) = p_1 * P_{in}(F_1, r, t) + p_2 * P_{in}(F_2, r, t) \quad (7)$$

$$P_{out}(F_0, r, t) = P_{out}(F_1, r, t) + P_{out}(F_2, r, t) \quad (8)$$

Here p_1 and p_2 are relative weights of F1 and F2. Combining the new inside and outside probability, the likelihood after merging F1 and F2 is:

$$LL'(S, T) = P_{in}(F_0, r, t) * P_{out}(F_0, r, t) + \sum_{x=2}^{k} P_{in}(F_x, r, t) * P_{out}(F_x, r, t) \quad (9)$$

The likelihood is approximated as:

$$\Delta = \frac{LL'(S,T)}{LL(S,T)} \quad (10)$$

Smoothing: Smoothing is another way of preventing overfitting. In Petrov et al., (2006), the probability of a production rule $P(F_x \to L_m R_n)$ is smoothed by interpolate it with the average value of probabilities over x.

$$P'(F_x \to L_m R_n) = (1 - \alpha) * P(F_x \to L_m R_n) + \alpha * \sum_x P(F_x \to L_m R_n) \quad (11)$$

3 Experiments

3.1 Setup

We divided the original Sinica Treebank data provided by the organizer into training and development set. To construct a representative dev--elopment set, we select every 10th sentence of the original data to add to the development set and use the rest of the sentences as the training set. The statistics of the training set and the development set are shown in table 1. "#word type" and "#tag type" denotes the number of different word forms and POS tags. "#non-terminals" denotes the number of non-terminal labels.

	Training set	development set
#sentence	55606	6179
#words	333996	37058
#word type	40593	11534
#tag type	101	68
#non-terminals	78	52
average length	6.01	6.00

Table1. Statistics of the training and development set

Though out this paper, we use the Berkeley parser[1] with the default settings to train all the parsing models. Parsing performance is evaluated using the evalb[2] program.

3.2 Experimental Results

The initial models are trained using our training data without any treatments. The parsing performances are listed in table 2.

From table 2, we can see that the best parsing performance in terms of F1 score is 78.16, and the best tagging accuracy is 91.60. These numbers are far below that achieved on the Penn Chinese Treebank (5.1) even the average length of the sentences in CTB is longer than the Sinica Treebank and we assume that the Sinica Treebank suffers more from data sparse problem. Interestingly, from table 2 we can see that the best parsing and tagging performance are both achieved at the 4-th split-merge round and after that parsing performance started to drop. This further confirms our assumption since for the WSJ Penn Treebank and the Penn Chinese Treebank, the best performance is achieved around the 6-th split-merge round.

#Split	Recall	Prec	F1	POS
1	71.67	74.63	73.12	90.78
2	75.61	77.04	76.32	91.28
3	77.56	77.96	77.76	91.60
4	78.28	78.04	78.16	91.58
5	77.50	76.89	77.19	91.00
6	76.88	76.15	76.51	90.05

Table 2. Parsing performance on the development set. #Split is the number of split-merge round

One should note that we do not argue the parsing performance of the Sinica Treebank and CTB are directly comparable. However, we do believe that the difference between the statistics of the two Treebanks helps to identify some difficulties in parsing the Sinica Treebank.

By comparing the statistics between the training set in this work and the training set of CTB, we found that the CTB contains more words, totally 536806 words, while less different word forms, 36922 word forms. Moreover, CTB only contains 42 different POS tags which is less than a half of the POS tags of the Sinica Treebank. These numbers demonstrate that parameters are more sufficiently estimated on CTB than on the Sinica Treebank.

By inspecting the detail tree structures and labels in the Sinica Treebank, we found that the Sinica Treebank annotation is more fine-grained compare with that of CTB. For POS tags, all words are divided into 8 basic categories including nouns, verbs, prepositions... Each category contains several sub-classes. For nouns, person names are annotated as Nb and organizations are annotated as Nc while in CTB, these two types of nouns are all tagged as NR. Moreover, some of the sub-classes are further distinguished with suffix such as VC[+NEG]. Non-terminals are annotated in a similar manner. In Sinica Treebank, all non-terminals belong to one of the 7 basic classes including noun phrase, verb phrase, preposition phrase... Each of the class contains several sub-classes which might be further distinguished by some suffixes.

The Sinica Treebank annotation does make its labels carry more information. However, the data sparse problem caused by the fine grained annotation prevents the Berkeley parser from learning a high performance model. To examine the effect of decreasing the number of label types on parsing performance, we carried on another two experiments. In the first experiment, we removed all suffixes from the POS tags and non-terminal labels of Sinica Treebank. For example, removing suffix from V_11 yields V and removing suffix from VC[+NEG] yields VC. After removing suffixes, the number of different POS tags decreased to 55. For the second experiments, in addition to remove all suffixes, we also maps all non-terminal labels to one of the seven phrase labels including NP, VP, GP, PP, XP, DM and S. These labels are used to measure parsing performance by the official backoff task evaluation metrics. The mapping procedure is conducted according to the first letter of the non-terminal label of the Sinica Treebank. That is, non-terminal labels with the first letter 'N' are all mapped to NP and labels with first letter 'V' are all mapped to VP …

[1] http://code.google.com/p/berkeleyparser/
[2] http://nlp.cs.nyu.edu/evalb/

Models	Prec	Recall	F1	POS
RAW	78.41	78.19	78.30	91.58
RMS	78.65	78.66	78.66	91.59
RMSM	75.77	75.62	75.69	89.97

Table 3. Parsing performance with different label set

Parsing performances are shown in table 3. "RAW" denotes the performance achieved on Sinica Treebank without any treatment. "RMS" denotes parsing performance achieved when label suffixes are removed. "RMSM" denotes parsing performance when both label suffixes are removed and non-terminals are mapped. For these settings, the best parsing performances in terms of F1 score are all achieved on the 4-th split-merge round and we omit the performance achieved on other rounds.

From table 3, we can see that on the one hand, 'RMS' improves parsing performance about 0.35 F1 points. This demonstrates that removing suffix to reduce the number of POS tag and non-terminal labels does to some degree helpful. On the other hand, aggressively maps non-terminal labels to only seven basic phrase labels hurts parsing performance dramatically.

Here, one may argue that these performances are not directly comparable since the gold development set for each setting are not annotated with the same label set. That is, scores for "RAW" setting is calculated against the development set without any treatment while scores for "RMS" setting is calculated against the development set which non-terminal labels' suffix are removed. For "RMSM", the gold development set only contains seven basic phrase labels. To handle this issue, we also mapped the parsing results of "RAW" and "RMS" to the seven basic phrase labels and the performance are listed in table 4. We see that "RMS_B" still yields the best performance.

The last issue we examine is tagging accuracy on parsing performance. To see this, we use the model trained with "RMS" setting to parse the development set where sentences are assigned with gold standard POS tags. The result is that parsing precision, recall and F1 boosted to 84.95, 84.44 and 84.69, respectively. These results illus trate that improving tagging accuracy can significantly boosting parsing performance on the Sinica Treebank. By inspecting the parsing errors which also evolve at least one tagging error, we found that one of the major sources of parsing errors is caused by Ncd-Ng ambiguity. Both the

Models	Prec	Recall	F1	POS
RAW_B	79.26	79.00	79.13	91.58
RMS_B	79.47	79.48	79.48	91.59
RMSM	75.77	75.62	75.69	89.97

Table 4. Parsing performance where non-terminal labels of the guess trees of "RAW" and "RMS" are mapped to seven basic phrase labels

two POS tags denote position information such as 外/'outside', 中/'in'. For example 校外/'outside the school', 庭院中/'in the yard'. However, the two tags show quite different syntactic behavior. Ng always coupled with NP or VP and they together forms a GP while Ncd always comes after a NP or a sequence of nouns to form another NP as shown in Figure1.

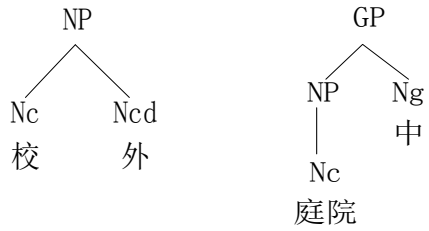

Figure1. Different syntactic structures between Ncd and Ng

Another major source of errors comes from noun-verb ambiguity which is also one of the most difficulty issues for tagging and parsing simplified Chinese. Such tagging error would results in a NP incorrectly analyzed as a VP and vice versa.

4 Conclusion

In this paper we analyze the difficulties in parsing the Sinica Treebank. We also examined the effect of tagging errors on parsing performance. We show that the fine-grained annotation schema of the Sinica Treebank is one major factor that prevents high parsing performance. In particular, the annotation schema leads to severe data sparse problem which makes the model parameters cannot be sufficiently estimated.

References

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL'05*, p. 173–180.

M. Candito, B. Crabb é and D. Seddah. On statistical parsing of French with supervised and semi-supervised strategies. 2009. In *EACL' 09*, p. 49-57

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, U. of Pennsylva

D. Guldea. 2001. Corpus variation and parser performance. In *EMNLP '2001*, p. 167-202

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. *ACL '03*, p. 423–430.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *ACL '05*, p. 75–82.

P. Slav and D. Klein. 2007. Improved Inference for Unlexicalized Parsing. In *NAACL' 2007*, p. 404-411

P. Slav, B. Leon, T. Romain and D. Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. *In COLING' 2006*, p. 433-440