A Simplified Chinese Parser with Factored Model

Qiuping Huang
Natural Language Processing & Portuguese-Chinese Machine Translation
Department of Computer and Information Science
University of Macau
michellehuang718@gmail.com

Liangye He
Natural Language Processing & Portuguese-Chinese Machine Translation
Department of Computer and Information Science
University of Macau
wutianshui0515@gmail.com

Derek F.Wong
Natural Language Processing & Portuguese-Chinese Machine Translation
Department of Computer and Information Science
University of Macau
derekfw@umac.mo

Lidia S.Chao
Natural Language Processing & Portuguese-Chinese Machine Translation
Department of Computer and Information Science
University of Macau
lidiasc@umac.mo

Abstract

This paper presents our work for participation in the 2012 CIPS-ParsEval shared task of Simplified Chinese parsing. We adopt a factored model to parse the Simplified Chinese. The factored model is one kind of combined structure between PCFG structure and dependency structure. It mainly uses an extremely effective A* parsing algorithm which enables to get a more optimal solution. Throughout this paper, we use TCT Treebank as experimental data. TCT mainly consists of binary trees, with a few single-branch trees. The final experiment result demonstrates that the head propagation table improves the parsing performance. Finally, we describe the implementation of the system we used as well as analyze our experiment result SC_F1 from 43% up to 63% and the LC_F1 is about 92% we have achieved.

1 Introduction

Parsing is an important and fundamental task in natural language processing. In recent years, Chinese parsing has received a great deal of attention, and lots of researchers have presented many of Chinese parsing models (Collins, 1999; Klein and Manning, 2003; Charniak and Johnson, 2005; Petrov, 2006). Nevertheless, the factored model is presented as a novel parsing model, which provides conceptually concise, straightforward opportunities for separately improving the component models (Klein and Manning, 2002).

With the efforts of many researchers, natural language processing makes a remarkable improvement and the syntactic analysis results can be directly used for machine translation, automatic question and answering and information extraction. However, most researches on parsing concentrating on English, and its parsing system has achieved quite a good performance. Thus the Chinese parsing is still a huge challenge in Chinese information processing.

Parsing is the thesis that analyzes the word's grammatical function in the sentence, and it also is a data driven process, its performance is determined by the amount of data in a Treebank on which a parser is trained (Song and Kit, 2009). Although much more multilingual parsing models have been presented, the data for English is still much more than any other languages that have been available so far. For this reason, most researches on parsing focus on English. If we directly apply any existing parser trained on an English Treebank for Chinese sentences, we cannot get a good parsing. However, the

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 188–193,
Tianjin, China, 20-21 DEC. 2012

Vertical Order	Horizontal Markov Order			
	$h = 0$	$h = 1$	$h = 2$	$h = \infty$
$v = 1$	$p(H\|L)$	$p(H\|L, M_k)$	$p(H\|L, M_k M_{k+1})$	$p(H\|L, M_1, M_2 \dots M_x)$
$v = 2$	$p(H\|L, P)$	$p(H\|L, P, M_k)$	$p(H\|L, P, M_k M_{k+1})$	$p(H\|L, P, M_1, M_2 \dots M_x)$
$v = 3$	$p(H\|L, P, G)$	$p(H\|L, P, G, M_k)$	$p(H\|L, P, G, M_k M_{k+1})$	$p(H\|L, P, G, M_1, M_2 \dots M_x)$

Table 1: Markovization and corresponding statistical model

methodology of parsing can be highly applicable. Even for those corpora with different annotation format, there still has a well-performed parser to fit the specific structure for the data. In this work, we adopt an existing powerful parser, Stanford parser (Klein and Manning, 2003), which has shown its effectiveness in English. We make the necessary modifications for parsing Chinese and apply it to the shared task.

In this evaluation, we use TCT Treebank as the developing and experimental data. The Treebank uses an annotation scheme with double-tagging (Zhou, 2004). Under this scheme, every sentence is annotated with a complete parse tree, where each non-terminal constituent is assigned with two tags, the syntactic constituent tag and the grammatical relation tag, which also is a new annotation scheme that differs from with head constituents in previous TCT version. In order to fit to this annotation of TCT, we use the unlexicalized model to do the PCFG parsing and use CKY-based decoder in the Stanford parser. Finally we mainly use TregEx (Levy, 2006), which is a useful tool to visualize and query syntactic structures, to generate a head propagation table applying to the factored model in order to improve the performance.

In the next section, we will present the details of our approach. The experiment results and analysis are presented in section 3. The last section is the conclusion and further work.

2 Parsing Model

2.1 Stanford Factored Model

The Stanford parser, precisely, the highly optimized factored model (Klein and Manning, 2003) has been employed to perform our experiment. The factored model is the combination of unlexicalized PCFG model and dependency model. To our knowledge, the unlexicalized model did not encode word information and the dependency model can be viewed as postprocessing in the Stanford factored model. The factored model can be seen as $P(T, D) = p(T)p(D)$, Where T means the plain phrase-structure tree and D is dependency tree. In this view, the factored model is built by two sub-models.

The Stanford unlexicalized PCFG model makes horizontal and vertical grammar markovizations to solve two deficiencies of raw grammar: coarse category symbols and the unknown testing rules. Coarse category symbols make too strong independent assumptions; while unknown testing rules often get underestimated probabilities. Assumed that h stands for horizontal markovization order, v stands for vertical markovization order, and every grammar rules are in this type:

$$L \rightarrow M_1 \dots M_i H M_{i+1} \dots M_x$$

In this rule, L is the left-hand-side, H is the head word in the right-hand-side, M_x stands for the modifiers. P indicates parent nodes and G indicates grandparent nodes (Klein and Manning, 2003). Table 1 gives the unlexicalized parsing models corresponding to different horizontal and vertical orders.

The dependency models $p(D)$ is a pair $< h, a >$ of a head and argument, which are words in a sentence. A dependency structure D over a sentence is a set of dependencies (arrows) which form a planar, acyclic graph rooted at the special symbol *ROOT*, and in which each word in sentence appears as an argument exactly once (Klein and Manning, 2004). The arrow connects a head with a dependent, and the head $< h, a >$ of a constituent is generated by the head propagation table. The CKY algorithm is used in dependency parsing.

Actually, the factored model reaches to the efficient by factoring the two sub-models and simplified both. There is a brief top-level procedure described in (Klein and Manning, 2002).

1. Extract the PCFG sub-model and set up the PCFG parser.
2. Use the PCFG parser to find outside scores $\alpha_{PCFG}(e)$ for each edge.
3. Extract the dependency sub-model and set up the dependency parser.
4. Use the dependency parser to find outside scores $\alpha_{DEP}(e)$ for each edge.

Parent Node	Child Node	Frequency
ap	*a*	19
	ap	13
	pp	8
	d	7
	dD	7
	vp	5
	aD	3

Table 2: The classification and frequency of *ap* node

Parent	Direction	Priority List
np	right	*n, np,vN,nP, mp, v, vp, rN, nR, m, sp, t, rNP, dj*
vp	left	*vp, v, n, tp, sp, vM, a, ap, p, pp, t*
ap	left	*a, ap, aD, d, dD,vp*
bp	left	*b, u*
dj	left	*vp, dj, np, n,b*
dlc	right	*dlc, l, np*
dp	right	*uJDI, dN,d*
fj	left	*fj-RT, fj*
mp	left	*qN, mp, m, tp, mbar-XX*
pp	left	*np,sp, n, tp, rN, pp, v, a, f*
sp	right	*f, n, nS, s, sp, np*
tp	right	*qT, nT, f, tp, n,np, m*
yj	right	*yj-RT*
jq	left	*jq, zj-XX*

Table 3: The head propagation table used in Simplified Chinese parsing

5. Combine PCFG and dependency sub-models into the lexicalized model.
6. Form the combined outside estimate $a(e) = \alpha_{PCFG}(e) + \alpha_{DEP}(e)$.
7. Use the lexicalized A* parser, with $a(e)$ as an A* estimate of $\alpha(e)$.

2.2 Head Propagation Table

It is worth mentioning that the headword information does not reflect on the parsed syntax tree for a given sentence in the corpus. In the case of dependency model, Stanford model mainly uses constituency structure to extract dependency grammar. On this hand, the headword information plays an important role. The parser needs to pick out the head child in the internal rules with the head propagation table. Besides, the Stanford factored model also is the combination of unlexicalized PCFG models and lexicalized models, it has to encode the lexicalized information in each non-terminal node. Likewise, the lexicalized parser uses the head propagation table as well. However, the newest TCT corpus does not contain the head word information. To this end, we define a specific head propagation table using the TregEx tool after classifying the grammar rules and counting the frequency of some related tags. Which differs from the work of (Magerman, 1995) and (Collins, 1999) that the rules of head finding are defined based on linguistic knowledge. There are three steps to generate the head propagation table. Firstly, we extract all the grammar rules from the TCT corpus, and then classify the rules according to their parent nodes. Secondly, we calculate the frequency of each sort of child node that have the same parent node, then select the higher frequency child nodes as the candidate head word. For example, under the *ap* (adjective phrase) node, we get some relatively high frequency child nodes by counting showed in the table 2. Thirdly, we search the matched sub-trees that the candidate head is the real head in the TCT Treebank by using the TregEx specified pattern (Levy, 2006). Finally, through the distribution of the amount of the matched tree fragment, we generate the head propagation table and every child node is assigned with a priority score and direction. The

generation of direction (left or right) is the combination of linguistic knowledge and experiment results. Table 3 gives the head propagation table used in our Simplified Chinese parsing. In the Stanford parser, there is an existed class of *LeftHeadFiner* which defaults the leftmost one is the head word. Similarly, we create a class of *RightHeadFinder* which defaults the rightmost one is the head word. In our task, we have used leftmost, rightmost, and the generated head propagation table to do three group experiments respectively. The experiment proved that after the head propagation table imported which indeed improves the result exceeding the other two experiments based on the same settings on the parser.

3 Experiment and Analysis

3.1 Data Set

In this work, all of news and academic articles annotated in the TCT version 1.0 (Zhou, 2004) are selected as the basic training data for the evaluation. 1,000 sentences extracted from the TCT-2010 version can be used as the basic test data. The Treebank uses a double-tagging annotation scheme. For example: (*zj-XX* (*fj-LS* (*dj* (*nP* 江泽民) (*v* 指出)) (*dj-RT* (*wP* ，) (*dj* (*vp* (*v* 搞好) (*np* (*n* 物价) (*n* 工作))) (*vp* (*dD* 极) (*vp* (*v* 为) (*a* 重要)))))) (*wE* 。)). In this sentence, *zj, dj, np,* etc. are the syntactic tags and *LS, RT* are grammatical relation tags. These two tag sets consist of 16 and 31 different tags respectively, which is a new annotation scheme with double-tagging that differing from with head constituents in previous version of TCT corpus. In addition, we have 10 different scale official released training data sets from TCT, but the latter data set has included the former data set. It is a cumulative manner. For example, the set 1 (means D_1) has 1,755 sentences , yet the set 2 has 3,512 sentences in all which includes all sentences of set 1. The any other data sets are generated according to the same idea. There are 17,558 sentences and about 480,000 Chinese words in the biggest official released training data set. In the corpus, every sentence contains 5 words at least and some sentences are more than 100 words. The more syntactic relation exists in the long sentence, the more difficulties exist in these complex sentences when parsing. In order to evaluate the effectiveness on the different scales of the training data for parser performance, we extract 90% data to training and 10% data for testing from 10 training data sets mentioned before, so there are 10 different training data sets and testing data sets. It is worth noting that the testing sets are also cumulative.

Furthermore, in order to use the Stanford parser, we need to transform format of the corpus that parentheses are added to delimiter the boundaries of sentences. Simultaneously, we create a Simplified Chinese package to do the parsing. This package mainly contains head finding rules, and some tuning of parser option for the TCT corpus.

3.2 Results and Analysis

The evaluation metrics used in 2012 CIPSParsEval shared task are shown in following:

$$Precision = \frac{\#\ of\ correct\ constituents\ in\ proposed\ parse}{\#\ of\ constituents\ in\ proposed\ parse}$$

$$Recall = \frac{\#\ of\ correct\ constituents\ in\ proposed\ parse}{\#\ of\ constituents\ in\ standard\ parse}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

There are two evaluation results in this shared task. One is the syntactic category (SC), the other is labeled constituent (LC).

As we mentioned before, we use cumulative manners to train 10 different training models. Table 4 gives the results which use the raw Treebank based on the default Chinese training setting on Stanford parser. This is an original model in our experiment. Table 5 shows the best results among three group experiments by importing three classes respectively. The first is the leftmost which always selects the leftmost as the headword (=1 in Table 5). The second is the rightmost which always selects the rightmost as the headword (=2 in Table 5) and the third is the head propagation table (=3 in Table 5). From the result, we can see that after the Simplified Chinese package and the head propagation table imported, we got the best PARSEVAL LC_F1 is about 92% and SC_F1 is close to 63% corresponding to $v = 2, h = \infty$. The table 6 shows the results of 10 different scales of the training data set in our adapted model by importing the head propagation table. We can see that with the more training data in a certain range, the model is more robust from 3 to 9 different scale data sets. However, tenth set declines slightly. There may be some reasons for the result. One, there are some unknown words appearing in the tenth set and cannot be recognized. Two, much more long sentences with more syntactic relation can not be parsed well in this data set. Three, the training data reaches an extreme point in the ninth set,

with the more data, the more ambiguities when selecting the grammar rules.

Data	LC_F1	SC_F1
D_1	85.12	38.42
$D_2 \supseteq D_1$	84.15	38.74
$D_3 \supseteq D_2$	86.52	41.03
$D_4 \supseteq D_3$	87.66	41.14
$D_5 \supseteq D_4$	88.61	41.39
$D_6 \supseteq D_5$	89.02	41.84
$D_7 \supseteq D_6$	89.51	42.50
$D_8 \supseteq D_7$	89.79	42.54
$D_9 \supseteq D_8$	90.20	42.81
$D_{10} \supseteq D_9$	90.04	42.26

Table 4: The parsing results based on the original model trained on different scales of training data

Experiment	LC_F	SC_F
1	91.79	59.80
2	91.80	60.00
3	91.88	62.81

Table 5: The best results among three groups of experiment on the adapted model

Data	LC_F	SC_F
D_1	90.49	61.26
$D_2 \supseteq D_1$	89.05	61.09
$D_3 \supseteq D_2$	89.56	60.37
$D_4 \supseteq D_3$	91.13	61.60
$D_5 \supseteq D_4$	90.98	61.71
$D_6 \supseteq D_5$	91.18	62.13
$D_7 \supseteq D_6$	91.47	62.60
$D_8 \supseteq D_7$	91.68	62.78
$D_9 \supseteq D_8$	91.88	62.81
$D_{10} \supseteq D_9$	91.88	62.69

Table 6: The parsing results of the adapted model trained on different scales of training data

4 Conclusion and Future Work

We participate in the parsing subtask in CIPS-Parseval 2012. We use the factored model of Stanford parser to tackle the parsing. The framework of factored model is conceptually simple and can be easily extended in some ways that other parser models have been. Besides, we mainly use the TregEx searching Treebank tool and counting manner to generate the head propagation table, though it makes sense to the parsing result, we still hope to find a better way to extend its feasibility and not just used for Simplified Chinese. Whether we can create the head table automatically based on machine learning. Perhaps this is a thought-provoking question in future research. However, there are some improvements we can make. At first, we can further study the double-tagging annotation scheme in TCT Treebank in order to do the tag splitting as done on English Treebank (Klein and Manning, 2003). Because the tag splitting is another important feature of Stanford parser. In addition, the head constituent recognition is the key problem, we hope a breakthrough in this problem.

Acknowledgments

This work was partially supported by the Research Committee of University of Macau under grant UL019B/09-Y3/EEE/LYP01/FST, and also supported by Science and Technology Development Fund of Macau under grant 057/2009/A2.

References

Collins, M. (1999). Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4): 589-637.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 173–180.

Huang, L. (2008). Forest reranking: Discriminative parsing with non-local features. *Proceedings of ACL-08: HLT*, 586–594.

Klein, D. and Manning, C. D. (2002). Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, *15*(2003), 3–10.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 423–430.

Klein, D. and Manning, C. D. (2003). Factored A* Search for Models over Sequences and Trees. *Proceedings of the International Joint Conference on Artificial Intelligence*, *18*, 1246–1251.

Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 478.

Levy, R. and Andrew, G. (2006). Tregex and Tsurgeon: tools for querying and manipulating tree data structures. *Proceedings of the fifth international conference on Language Resources and Evaluation*, 2231–2234.

Magerman, D. M. (1995). Statistical decision-tree models for parsing. *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 276–283.

Petrov, S., Barrett, L., Thibaux, R. and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 433–440.

Song, Y. and Kit, C. (2009). PCFG parsing with CRF tagging for head recognition. *Proceedings of CIPS-ParsEval*, 133–137.

Zhou Q. 2004. Annotation Scheme for Chinese treebank. *Journal of Chinese Information Processing*, 18(4):1-8.