# Parsing TCT with a Coarse-to-fine Approach

**Li Dongchen**
Key Laboratory of Machine Perception and Intelligence,
Speech and Hearing Research Center
Peking University, China
lidc@cis.pku.edu.cn

**Wu Xihong**
Key Laboratory of Machine Perception and Intelligence,
Speech and Hearing Research Center
Peking University, China
wxh@cis.pku.edu.cn

## Abstract

A key observation is that concept compound constituent labels are detrimental to parsing performance. We use a PCFG parsing algorithm that uses a multilevel coarse-to-fine scheme. Our approach requires a sequence of nested partitions or equivalence classes of the PCFG nonterminals, where the nonterminals of each PCFG are clusters of nonterminals of the finer PCFG. We use the results of parsing at a coarser level to prune the next finer level. The coarse-to-fine method use hierarchical projections for incremental pruning. We present experiments which show that parsing with hierarchical state-splitting is fast and accurate on Tsinghua Chinese Treebank. In addition, we propose a multiple-model method that adds concept compound labels to the output of the simple PCFG model and gains higher bracketing recall from the simple model. This scheme can be implemented by training two models on different labeling styles.

## 1 Introduction

The peculiarity of the annotation of this released edition of TCT is that the tree structure is very compact, where there are no unary productions except root nodes and leaf nodes.

A major observation is that parser on treebank with concept compound constituent labels performs worse than without concept compound constituent. The average crossing is 4% lower in presence of concept compound constituent labels.

Since all phrases have a clausal and phrasal constituent label, while only a fraction have concept compound constituent label. We can regard a phrase label with both clausal and phrasal constituent label and concept compound constituent label as a subsymbol of the clausal and phrasal constituent label merely.

The coarse categories in these grammars can be regarded as clusters or equivalence classes of the fine nonterminal categories. We require that the partition of the nonterminals defined by the equivalence classes at finer level be a refinement of the partition defined at coarser level. This means that each nonterminal category at finer level is mapped to a unique nonterminal category at coarser level (although in general the mapping is many to one, i.e., each nonterminal category at coarser level corresponds to several nonterminal categories at finer level). We use the correspondence between categories at different levels to prune possible constituents. A constituent is considered at finer level only if the corresponding constituent at coarser level has a probability exceeding some threshold. Parsing with hierarchical grammar leads to considerable efficiency improvements.

Treebank parsing comprises two problems: learning, in which we must select a model given a treebank, and inference, in which we must select a parse for a sentence given the learned model. Previous work has shown that high-quality unlexicalized PCFGs can be learned from a treebank, either by manual annotation (Klein and Manning, 2003) or automatic state splitting (Matsuzaki et al., 2005; Petrov et al., 2006). In particular, we demonstrated in Petrov et al. (2006) that a hierarchically split PCFG could exceed the accuracy of lexicalized PCFGs (Collins, 1999; Charniak and Johnson, 2005).

We adopted here a multilevel coarse-to-fine PCFG parsing algorithm as in Charniak (2006) and Petrov (2007). The multilevel coarse-to-fine PCFG parsing algorithm reduces the complexity of the search involved in finding the best parse and attempts to constrain the fine parsing space



*Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 194–198,
Tianjin, China, 20-21 DEC. 2012

to the coarse parsing space. It defines a sequence of increasingly more complex PCFGs. Charniak (2006) has demonstrated that coarse PCFG identified the locations of correct constituents of the parse tree (the "gold constituents") with high recall.

## 2 Experiment Observation

We have parsed with three different annotation setups. First, we train our model our model with only phrasal labels, and evaluate the precision and recall on only the phrasal labels. Second, we train our model with full labels, and evaluate the precision and recall on only the phrasal labels. Third，we train the model with full labels, and evaluate the precision and recall on full labels.

Take a concrete example, we show two parsing output with different annotations as below:

The input sentence is:

"之后， 北京 一轻 总公司 根据 市政府 的 决定， 在 市 国有 资产 管理局 的 具体 指导 下， 经过 3 个 月 的 紧张 工作， 完成 了 公司 国有 资产 的 清查 、 重估 工作 。"

Parsing output with only phrasal constituent labels:

"( (zj (dj (t 之后) (dj (wP ， ) (dj (np (np (nS 北京) (n 一轻)) (n 总公司)) (vp (pp (p 根据) (np (np (n 市政府) (uJDE 的)) (n 决定))) (vp (wP ， ) (vp (pp (p 在) (sp (np (np (np (np (n 市) (np (b 国有) (n 资产))) (n 管理局)) (uJDE 的)) (np (a 具体) (vN 指导))) (f 下))) (vp (wP ， ) (vp (pp (p 经过) (np (np (tp (mp (m ３) (qN 个)) (qT 月)) (uJDE 的)) (np (a 紧张) (n 工作)))) (vp (wP ， ) (vp (vp (v 完成) (uA 了)) (np (np (np (n 公司) (np (b 国有) (n 资产))) (uJDE 的)) (np (np (n 清查) (np (wD 、 ) (n 重估))) (n 工作)))))))))))) (wE 。)) )"

Parsing output with full labels:

"( (zj_XX (fj (f 之后) (fj_RT (wP ， ) (fj_LG (dj (np (nS 北京) (np (n 一轻) (n 总公司))) (vp (pp (p 根据) (np (np (n 市政府) (uJDE 的)) (n 决定))) (vp_RT (wP ， ) (vp (pp (p 在) (sp (np (np (np (n 市) (np (np (b 国有) (n 资产)) (n 管理局))) (uJDE 的)) (np (a 具体) (vN 指导))) (f 下))) (vp_RT (wP ， ) (vp (v 经过) (np (np (tp (mp (m ３) (qN 个)) (qT 月)) (uJDE 的)) (np (a 紧张) (n 工作))))))))) (vp_RT (wP ， ) (vp (vp (v 完成) (uA 了)) (np (np (np (n 公司) (np (b 国有) (n 资产))) (uJDE 的)) (np (np_LH (vN 清查) (np_RT (wD 、 ) (vN 重估))) (n 工作)))))))) (wE 。)) )"

The gold parse tree is as follows:

"( (zj (dj (f 之后) (dj (wP ， ) (dj (np (np (nS 北京) (n 一轻)) (n 总公司)) (vp (pp (p 根据) (np (np (n 市政府) (uJDE 的)) (n 决定))) (vp (wP ， ) (vp (pp (p 在) (sp (np (np (np (n 市) (np (b 国有) (np (n 资产) (n 管理局)))) (uJDE 的)) (vp (aD 具体) (v 指导))) (f 下))) (vp (wP ， ) (vp (pp (p 经过) (np (np (tp (mp (m ３) (qN 个)) (qT 月)) (uJDE 的)) (np (a 紧张) (n 工作)))) (vp (wP ， ) (vp (vp (v 完成) (uA 了)) (np (np (np (n 公司) (np (b 国有) (n 资产))) (uJDE 的)) (np (np (vN 清查) (np (wD 、 ) (vN 重估))) (n 工作)))))))))))) (wE 。)))"

In the former parsing result, not only the phrasal constituent tags are labels more accurately, its syntactic structures are segmented more reasonably.

The parsing performances metrics convinced that the concept compound is detrimental to the parser performance even we only evaluate the phrasal constituent labels' precision and recall.

Furthermore, we compare the metrics of exact match, average crossing, no crossing and 2 or less crossing, which show that the higher accuracy gained by stripping the concept compound labels lies in both its more accurate bracketing and tagging ability.

## 3 Previous Researches

Coarse-to-fine search is an idea that has appeared several times in the literature of computational linguistics and related areas. Maxwell and Kaplan (1993) extracted CFG automatically from a more detailed unification grammar and used it to identify the possible locations of constituents in the more detailed parses of the sentence. They use their covering CFG to prune the search of their unification grammar parser in essentially the same manner as we do here, and demonstrate significant performance improvements by using their coarse-to-fine approach.

Geman and Kochanek (2001) laid out the basic theory of coarse-to-fine approximations and dynamic programming in a stochastic framework. They describes the multilevel dynamic programming algorithm needed for coarse-to-fine analysis (which they apply to decoding rather than parsing), and show how to perform exact coarse-to-fine computation, rather than the heuristic search we perform here.

Goodman (1997)'s parser is a two-stage coarse to fine parser. The second stage is a standard tree-bank parser while the first stage is a regular-expression approximation of the gram-

mar. Again, the second stage is constrained by the parses found in the first stage. Neither stage is smoothed.

The parser of Charniak (2000) is also a two-stage coarse to fine model, where the first stage is a smoothed Markov grammar (it uses up to three previous constituents as context), and the second stage is a lexicalized Markov grammar with extra annotations about parents and grandparents. The second stage explores all of the constituents not pruned out after the first stage. Related approaches are used in Hall (2004) and Charniak and Johnson (2005).

Klein and Manning (2003a) describe efficient A* for the most likely parse, where pruning is accomplished by using Equation 1 and a true upper bound on the outside probability. While their maximum is a looser estimate of the outside probability, it is an admissible heuristic and together with an A* search is guaranteed to find the best parse first. One question is if the guarantee is worth the extra search required by the looser estimate of the true outside probability.

Tsuruoka and Tsujii (2004) explore the framework developed in Klein and Manning (2003a), and seek ways to minimize the time required by the heap manipulations necessary in this scheme. They describe an iterative deepening algorithm that does not require a heap. They also speed computation by precomputing more accurate upper bounds on the outside probabilities of various kinds of constituents. They are able to reduce by half the number of constituents required to find the best parse (compared to CKY).

McDonald et al. (2005) have implemented a dependency parser with good accuracy (it is almost as good at dependency parsing as Charniak (2000)) and very impressive speed (it is about ten times faster than Collins (1997) and four times faster than Charniak (2000)). It achieves its speed in part because dependency parsing has a much lower grammar constant than does standard PCFG parsing — after all, there are no phrasal constituents to consider. The current paper can be thought of as a way to take the sting out of the grammar constant for PCFGs by parsing first with very few phrasal constituents and adding them only after most constituents have been pruned away.

## 4 Hierarchically Split PCFGs

We use a novel coarse-to-fine processing scheme for hierarchically split PCFGs. Our method considers the splitting history of the final grammar, projecting it onto its increasingly refined prior stages. For any projection of a grammar, we use techniques for infinite tree distributions (Corazza and Satta, 2006) and iterated fix point equations. We then parse with each refinement, in sequence, much along the lines of Charniak et al. (2006).

We consider PCFG grammars in a hierarchy fashion in Petrov et al. (2006). From the starting point of the raw treebank grammar, we iteratively refine the grammar in stages. The refined grammar is estimated using a variant of the forward-backward algorithm (Matsuzaki et al., 2005). After a splitting stage, many splits are rolled back based on (an approximation to) their likelihood gain. This procedure gives an ontogeny of grammars from the raw grammar to the final grammar. Empirically, the gains on the English Penn treebank level off after 6 rounds.

## 5 Coarse-to-Fine Search

When working with large grammars, it is standard to prune the search space in some way. In the case of lexicalized grammars, the unpruned chart often will not even fit in memory for long sentences. Several proven techniques exist. Collins (1999) combines a punctuation rule which eliminates many spans entirely, and then uses span-synchronous beams to prune in a bottom-up fashion. Charniak et al. (1998) introduces best-first parsing, in which a figure-of merit prioritizes agenda processing. Most relevant to our work is Charniak and Johnson (2005) which uses a pre-parse phase to rapidly parse with a very coarse, unlexicalized treebank grammar. Any item with sufficiently low posterior probability in the pre-parse triggers the pruning of its lexical variants in a subsequent full parse.

Charniak et al. (2006) introduces multi-level coarse-to-fine parsing, which extends the basic pre-parsing idea by adding more rounds of pruning. In their work, the extra pruning was with grammars even coarser than the raw treebank grammar, such as a grammar in which all nonterminals are collapsed. We propose a novel multi-stage coarse-to-fine method which is particularly natural for our hierarchically split grammar, but which is, in principle, applicable to any grammar.

Petrov et al. (2007) construct a sequence of increasingly refined grammars, reparsing with each refinement. They derive sequences of refinements and automatically tune the pruning thresholds on held-out data. Their hierarchical

coarse-to-fine parsing take the projection that collapses split symbols in finer round to their earlier identities in coarser round. The final state-split grammars G come, by their construction process, with an ontogeny of grammars where each grammar is a (partial) splitting of the preceding one.

## 6 Experimental Setup

We ran experiments on TCT. The training and test data set splits are described in Table below.

| Treebank | Train Dataset | Develop Dataset | Test Dataset |
|---|---|---|---|
| TCT(Qiang Zhou, 2004) | 16000 sentences | 800 sentences | 758 sentences |

Table 1. Experiment DataSet Setup

Tsinghua Chinese Treebank is a 1,000,000 words Chinese treebank covering a balanced collection of journalistic, literary, academic, and other documents.

## 7 Final Results

We took the final model and used it to parse the specified test set in the 3nd Chinese Parsing Evaluation which contains 1000 sentences, and achieved the best precision, recall and F-measure. We use the evaluation method released by CLP 2012.

| SC_F1 | ULC_P | ULC_R | ULC_F1 |
|---|---|---|---|
| 92.29% | 87.02% | 87.04% | 87.03% |

Table 2. Experiment Results of SC and ULC

| NoCross_P | LC_P | LC_R | LC_F1 |
|---|---|---|---|
| 87.02% | 77.29% | 77.32% | 77.30% |

Table 3. Experiment Results of LC

| LC_P | LC_R | LC_F1 |
|---|---|---|
| 76.35% | 76.20% | 76.27% |

Table 4. Experiment Results of Tot4

Where LR = label recall, LP = label precision, F1 = F-measure, EX = exact match, AC = average crossing, NC = no crossing, 2C = 2 or less crossing.

## 8 Another Relabeling Method

A major observation is that concept compound constituent labels are detrimental to parsing performance. Since clausal and phrasal constituent labels are obligatory, while concept compound constituent labels are optional, we can strip concept compound constituent labels and parse with only clausal and phrasal constituent labels. Experiments show that parsing performance without concept compound constituents labels, especially the bracketing precision is significantly superior to the one with concept compound constituents labels.

Therefore, parsing directly with full labels (both clausal and phrasal constituent labels and concept compound labels) is unwise. In this paper, we get the concept compound label by the parser with full label, but get the extra performance gain by the parser with only clausal and phrasal constituent labels.

## 9 Integration of Both Parser

Clausal and phrasal constituent labels distinguish constituent phrasal categories, and full label (phrasal constituent label together with compound constituent label) moves forward to distinguish constituent structures.

A parser trained on the trees with only phrasal constituent labels have higher bracketing accuracy and phrasal constituent labels tagging accuracy. While another step can label the decoded tree with concept compound tags, either by incorporating the concept compound labels from the output of a parser trained on full label, or by relabeling the concept compound labels with a maximum entropy model.

In order to get strength from the both the parser output with and without concept compound labels, we train parser on both trees with only phrasal constituents label and full label, then add the concept compound labels from the later parser to the phrasal constituent labels from the former parser.

The simple PCFG identified the locations of correct constituents of the parse tree (the “gold constituents”) with high precision and recall. Then we label the concept compound labels in corresponds to the complex PCFG.

## 10 Conclusion

We employ a novel parsing algorithm based upon the coarse-to-fine processing model. It takes

the unpruned constituents and specifying them in the next level of granularity.

The coarse-to-fine scheme allows fast, accurate parsing. For training, one needs only a raw context-free treebank, and for decoding one needs only a final grammar, along with coarsening maps.

In addition, we propose a delicate integration method based upon two independently trained parsing models with different tree annotation style. The final output gains the higher bracketing label precision and recall from simpler tree annotation style, and adding the concept compound labels form the more complex tree annotation model.

**Acknowledgements**

This research is supported in part by the National Basic Research Program of China (No.2013CB329304) and the Key Program of National Social Science Foundation of China (No. 12&ZD119).

## References

E. Charniak and M. Johnson. 2005. Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking. In ACL'05.

E. Charniak, S. Goldwater, and M. Johnson. 1998. Edge-based best-first chart parsing. 6th Wkshop on Very Large Corpora.

E. Charniak,M. Johnson, et al. 2006. Multi-level coarse-to-fine PCFG Parsing. In HLT-NAACL '06.

Petrov, S., and Klein, D. 2007. Improved inference for unlexicalized parsing. In HLT-NAACL '07.

Petrov, S.; Barrett, L.; Thibaux, R.; and Klein, D. 2006. Learning accurate, compact, and interpretable tree annotation. In ACL '06.

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

A. Corazza and G. Satta. 2006. Cross-entropy and estimation of probabilistic context-free grammars. In HLT-NAACL '06.

M. Dreyer and J. Eisner. 2006. Better informed training of latent syntactic features. In EMNLP '06, pages 317–326.

J. Finkel, C. Manning, and A. Ng. 2006. Solving the problem of cascading errors: approximate Bayesian inference for lingusitic annotation pipelines. In EMNLP '06.

J. Goodman. 1996. Parsing algorithms and metrics. ACL '96.

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

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

D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. In COLING-ACL'06.

M. Mohri and B. Roark. 2006. Probabilistic context-free grammar induction based on structural zeros. In HLT-NAACL '06.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In COLING-ACL '06, pages 443–440.

H. Sun and D. Jurafsky. 2004. Shallow semantic parsing of Chinese. In HLT-NAACL '04, pages 249–256.

Dan Klein and Chris Manning. 2003a. A* parsing: Fast exact viterbi parse selection. In Proceedings of HLT-NAACL'03.

N. Xue, F.-D. Chiou, and M. Palmer. Building a large scale annotated Chinese corpus. In COLING '02, 2002.

Qiang Zhou. Chinese Treebank Annotation Scheme. Journal of Chinese Information, 18(4), p1-8. (2004)

Qiang Zhou, Yuemei Li. Evaluation report of CIPS-ParsEval-2009. In Proc. of First Workshop on Chinese Syntactic Parsing Evaluation, Beijing China, Nov. 2009. pIII—XIII. (2009)

Qiang Zhou, Jingbo Zhu. Chinese Syntactic Parsing Evaluation. Proc. of CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP-2010), Beijing, August 2010, pp 286-295. (2010)