Trigram-Inspired Three-Level PCFG Model and Its Application in Parsing Classical Chinese Texts

Liang HUANG[1], Yinan PENG[1], Huan WANG[2], Zhengyu WU[1]

[1] Department of Computer Science, Shanghai Jiaotong University
No. 1954 Huashan Road, Shanghai
P.R. China 200030
{lhuang, ynpeng, neochinese}@sjtu.edu.cn

[2] Department of Chinese Literature and Linguistics, East China Normal University
No. 3663 North Zhongshan Road, Shanghai,
P.R. China 200062

Abstract The traditional Probabilistic Context-Free Grammar (PCFG) model based on two-level CFG rules is widely used for parsing natural languages, but generally the accuracies are far from satisfactory, mainly because pure context-free grammars are difficult to incorporate contextual information. So in this paper, we improve the grammar model to a *three-level PCFG* that is to some extent quite "context-sensitive". Noticing that CFGs expand a non-terminal *N* using a rule *r* regardless of *N*'s parent and siblings, Charniak [8] proposed a "pseudo context-sensitive" model that takes *N*'s parent into account. Now in our model, the expansion of a non-terminal *N* depends on not only *N*'s parent but also the rule used by the parent to generate *N*. By considering the parent's generation rule, we can explicitly obtain the siblings, i.e., the context of *N*, thus gaining much more context-sensitivity than the model of [8]. For its applications, we found it especially successful in parsing Classical Chinese texts, partly because this model is more suitable to those word-order sensitive languages than such inflectional languages as English or French. The new three-level parser is just slightly different from the parser for two-level PCFG, and only scarifies a little efficiency and simplicity, while attaining substantially higher accuracies in our preliminary experiment. Our three-level model is inspired by the trigram model and they share many similarities.

Key words Three-Level Probabilistic Context-Free Grammar, Context sensitivity, Grammar model, Classical Chinese processing, top-down parsing

1 Introduction

In this paper we will propose and analyze an innovative trigram-inspired *three-level PCFG model* that, by its intrinsic *context-sensitivity*, significantly out-performs the traditional two-level PCFG model in terms of parsing accuracies. The experiments are carried out on parsing Classical Chinese texts where we get a very promising result. In this section, we will first briefly review previous works on PCFG/PCSG, then provide the background of Classical Chinese processing, and finally give the outline of the rest of the paper.

Context-free grammars (CFGs) are good models for parsing natural languages in that by a small-sized set of generating rules, one can cover almost all of the language phenomena and develop an efficient parser. But its drawbacks are also obvious, as virtually all the natural languages are context-sensitive in nature. So generally, pure PCFG parsers make a lot of errors. Some techniques may help improve the accuracies by some predefined preferences [9], but the modifications are trivial, nearly nothing to do with the model's inherent context independence. Charniak, however, improved the PCFG model in [8] with a "pseudo context-sensitive" grammar (PCSG) where a non-terminal N expands using a rule r dependent on N's parent, say N^s, as shown in equation (1) and Figure 1.

$$\Pr(N \to \alpha \mid \rho(N) = N^s) \qquad \textbf{(1)}$$

Where $\rho(x)$ denotes the non-terminal that immediately dominates x – its *parent*.

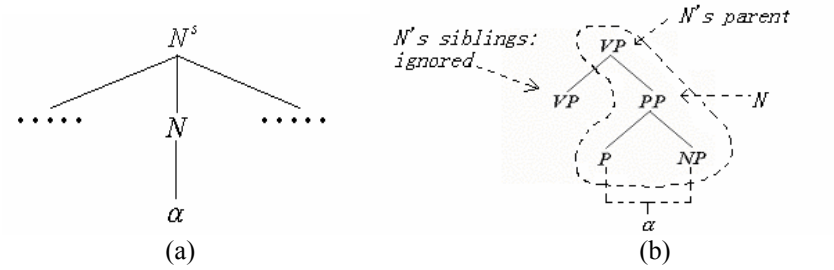


Fig. 1. (a) Charniak [8]'s pseudo CSG model, (b) an example: $\Pr(PP \to P\ NP \mid \rho(PP) = VP)$

But as shown in figure 1, Charniak's PCSG is still largely *context-free* because the non-terminal N's siblings are ignored. It is true that considering the parent does give some contextual information, but it does not explicitly provide N's left or right contexts. The parent N^s may contain several different rules to derive N, each of which differs in the siblings of N, as shown in figure 2.

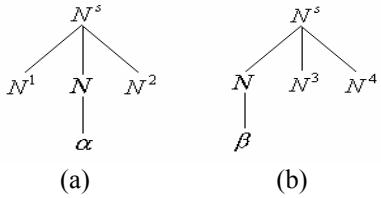


Fig. 2. Two different rules by the same parent to generate N, bringing different contexts of N, which is ignored by Charniak's model, that is, her model will not discriminate between these two parse trees.

In this paper, we devise a more context-sensitive model which takes both N's parent and the rule used by the parent to generate N into consideration when expanding the non-terminal N. The main advantage of this model is the discrimination of different contextual environment of N caused by the parent's different rules to generate N. In this way, by obtaining the parent of N and its rule to generate N, the siblings of N are *explicitly* known. Although our model is not theoretically context-sensitive, it substantially improves the parsing accuracies in our preliminary experiments on Classical Chinese texts, and unlike other techniques in the literature, our model increases only a little time complexity. It just mined more information from the same corpus (treebanks), just as trigrams versus bigrams. Our model will be presented in Section 3.

Another pioneering aspect of our work is the processing of Classical Chinese. As used by the Chinese people as the formal language from approximately 2000 B.C. to the early 20th century, Classical Chinese is essentially different from Modern Chinese, especially in syntax and morphology. While there has been a number of works on Modern Chinese Processing over the past decade [3, 6], Classical Chinese is largely neglected, mainly because of its obsolete and difficult grammar patterns. Huang et al., however, stated in [4] that Classical Chinese is even easier in terms of part-of-speech (POS) tagging, as there is almost no need of word segmentation, an inevitable obstacle in the processing of Modern Chinese texts [4].

Now in this paper, we move on to the parsing of Classical Chinese by PCFG model. Our parsing is a POS level parsing, not word level parsing. We use the same tagset as [4], and apply the forward-backward algorithm to obtain the context-dependent probabilities. Then for the PCFG model, we restrict it into binary/unary tree model which will simplify our programming. And we designed a set of context-free rules that could account for most, if not all, of the Classical Chinese grammatical phenomena encountered in our corpus (about 5000 sentences). In the first phase of our experiments, the results showed that pure two-level PCFG makes so many errors that the accuracy is even lower than 60%. A modification that takes predefined preferences into account could correct some of the errors and improve the accuracy to be about 80%. After analyzing the typical errors by the two-level PCFG model, we devised the three-level pseudo context-sensitive model that could get an accuracy of as high as 94.7%.

The rest of the paper is organized as follows. In Section 2, a tagset designed specially for Classical Chinese is introduced and the forward-backward algorithm for obtaining the context-dependent probabilities briefly discussed. In Section 3, we will briefly present the traditional two-level PCFG model, the syntactic tagset and CFG rule-set for Classical Chinese. The innovative three-level PCFG model is proposed in Section 4 and some features and comparisons with other grammar models are covered. The top-down parsing algorithm for the new model is designed to be both efficient and simple. We will analyze our preliminary experimental results in Section 5. A summary of the paper is given in Section 6.

2 Tagset and Context-Dependent Probabilities

Generally speaking, the design of tagset is very crucial to the accuracy and efficiency of tagging and parsing, and this was commonly neglected in the literature where many researchers use those famous corpora and their tagset as the standard test-beds. Still there should be a tradeoff between accuracy and efficiency. In the work by Huang et al. [4], a small-sized tagset for Classical Chinese is presented that is shown to be accurate in their POS tagging experiments. We will continue to use their tagset in this paper. We will also use a forward-backward algorithm to obtain the context-dependent probabilities.

2.1 Tagset

The tagset was designed with special interest not only to the lexical categories, but also the categories of components, namely *subcategories* a word may belong. For example, it discriminates adjectives into 4 subcategories like *Adjective as attributive*, etc. (See table 1). And several grammatical features should be reflected in the tagset. These discriminations and features turn out to be an important contributing factor of the accuracy in our parsing experiments.

Table 1. The tagset for Classical Chinese

Tag	Meaning	English meaning	Examples
n	名词	Noun	楚***人***有直躬
aa	形容词作定语	Adjective as attributive	***楚***人有直躬
aw	形容词作谓语	Adjective as verbal phrase	被甲者***少***也
ab	形容词作表语	Adjective as predicate	仲尼以为***孝***
ad	副词	Adverb	***必***禁无用
vi	不跟宾语的动词	Verb without object	知者不***惑***
vt	跟宾语的动词	Verb with object	今***修***文学
conj	连词	Conjunction	君子和***而***不同
yq	语气词	Exclamation	被甲者少***也***
prep	带宾语的介词	Preposition with object	应之***以***乱则凶

prepb	省略宾语的介词	Preposition with object omitted	仲尼***以***[之]为孝
num	数词	Number	虽有***十***黄帝
qpron	疑问代词	Wh-pronoun	则人***孰***不为也？
npron	名词性代词	Noun-pronoun	而人主兼礼***之***。
apron	形容词性代词	Adjective-pronoun	故明主用***其***力。
za	“之”作定语后置标志	*Special for Old Chinese*	乡人***之***善者
zj	“者”作名词性词尾	*Special for Old Chinese*	乡人之善***者***
zd	“之”作“的”	*Special for Old Chinese*	古***之***人不余欺。
fy	发语词	*Special for Old Chinese*	***夫***离法者罪。
conjad	副词性连词	Adverbial conjunction	用之***则***乱法。
Period	终止性标点		。；？！
Comma	停顿性标点		，、：

2.2 Tagging Algorithms

We apply the Hidden Markov Model (HMM) [1, 2] and the forward-backward algorithm [1] to obtain the context-dependent probabilities.

The problem is that all the estimations are based on bigram model, thus not accurate enough. And trigram model is difficult to be applied in the forward-backward algorithm. In our experiments, this algorithm does make a lot of errors. But fortunately however, the PCFG parser could correct these errors because the tagset is designed to be accurate enough, as is shown in our experiments.

Generally there are 2 types of HMM taggers for parsers, the trigram model and the bigram forward-backward model. Charniak suggested in [9] that the former is better for parsers. But the former only result in a sequence of most probable POS, in other words, it assigns only *one* POS tag for each word. Although the accuracy of trigram in [4] is as high as 97.6%, for a sentence of 10 words long, the possibility of all-correctness is as low as low as $(97.6\%)^{10} = 78.4\%$, and the single-tag scheme does not allow parsers to re-call the correct tags, as is often done in the forward-backward model. So in this paper we still apply the traditional bigram forward-backward algorithm. We suggest that a combination of trigram and forward-backward model would be the best choice, although no such attempt exists in the literature.

3 Two-Level PCFG Model and Classical Chinese Grammar

In this section we will first review the traditional two-level PCFG model and context-sensitive rules designed for Classical Chinese. Features of the rule-set will be discussed.

3.1 Two-Level PCFG Model

CFG: A context-free grammar (CFG) is a quadruple (V_N, V_T, S, R) where V_T is a set of terminals (POS tags), V_N is a set of non-terminals (syntactic tags), $S \in V_N$ is the start non-terminal, and *R* is the finite set of rules, which are pairs from $V_N \times V^+$, where V denotes $V_N \bigcup V_T$. A rule $< A, \alpha >$ is written in the form $A \to \alpha$, A is called the left hand side (LHS) and α the right hand side (RHS).

PCFG: A probabilistic context-free grammar (PCFG) is a quintuple (V_N, V_T, S, R, P), where (V_N, V_T, S, R) is a CFG and $P : R \mapsto (0,1]$ is a probability function such that $\forall N \in V_N : \sum_{\alpha: N \to \alpha \in R} P(N \to \alpha) = 1$

Rule Restriction: We restrict the CFG rules to be binary or unary rules, but ***NOT*** as strict as the Chomsky Normal Form (CNF). Each $R_i \in R$ could be in the following two forms only:

1. $R_i : N_j \to AB$
2. $R_i : N_j \to A$

where $N_j \in V_N$ and $A, B \in V$

The advantage of binary/unary rules lies in the simplicity of parsing algorithm, and will be discussed in Section 4.

The major difference between our model and CNF is that for unary rules, we do not require the right-hand-side to be terminals. And this enables us easier representation of the Classical Chinese language.

3.2 Rule-Set for Classical Chinese

An important advantage of PCFG is that it needs fewer rules and parameters. And according to our corpus, which is representative of Classical Chinese classics, only 100-150 rules would be sufficient. This is mainly because our rule set is linguistically sound. A summary of the set of rules is presented as follows.

Table 2. Our non-terminals (also called syntactic tagset, or constituent set)

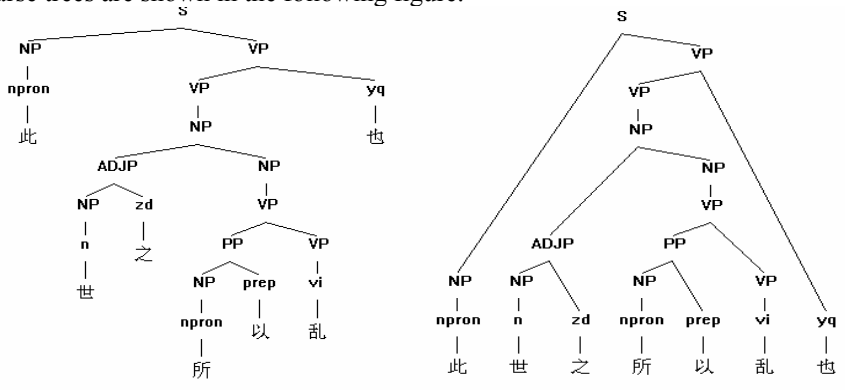
No.	Tags	Meaning	Examples
1	NP	Noun Phrase	***古之人***不余欺
2	VP	Verb Phrase	古之人***不余欺***
3	S	Sentence	***古之人不余欺***。
4	ADJP	Adjective Phrase	***楚***人有直躬
5	PP	Prepositional Phrase	应之***以乱***则凶
6	PADJP	Post-Adjective Phrase	乡人***之善者***
7	POSTADJP	The main part of PADJP	乡人之***善者***
8	PREDP	Predicate Phrase	仲尼以为***孝***。

A subset of frequently used rules is shown in the following table.

Table 3. A simple subset of PCFG Rules for Classical Chinese

1.	S ->	NP VP	; simple S/V	14.	NP ->	npron	
2.	S ->	VP	; S omitted	15.	NP ->	ADJP NP	
3.	S ->	VP NP	; S/V inversion	16.	NP ->	POSTADJP	
4.	S ->	ad S		17.	NP ->	VP	; V/O as NP
5.	VP ->	vi		18.	NP ->	fy NP	
6.	VP ->	vt NP	; simple V/O	19.	ADJP ->	aa	
7.	VP ->	NP vt	; V/O inversion	20.	ADJP ->	apron	
8.	VP ->	ad VP		21.	ADJP ->	NP zd	
9.	VP ->	PP VP	; prepositioned PP	22.	PP ->	prep NP	; P+NP
10.	VP ->	VP PP	; postpositioned PP	23.	PP ->	NP prep	; inversion
11.	VP ->	NP	; NP as VP	24.	PP ->	prepb	; object omitted
12.	VP ->	VP yq		25.	PP ->	NP	; prep. omitted
13.	NP ->	n		26.	POSTADJP->	VP zj	

Examples of parse trees are shown in the following figure.



|(a) (b)

Fig. 3. the parse trees of the sentence *此世之所以乱也*。(a) top-down parsing (b) bottom-up parsing

3.3 Features of Classical Chinese Grammar Rules

As mentioned before, the grammar of Classical Chinese is entirely different from that of English, so a few special features must be studied. And these features are difficult for the parser and the two-level model does make a lot of errors on these features. And our three-level parser is inspired by these typical errors.

From the rule-set, the reader might find that two special grammatical structures is very common in Classical Chinese:

1. Inversion: subject/verb inversion (rule 3), preposition/object inversion (rule 23).
2. Omission: Subject omitted (rule 2), preposition's object omitted (rule 24), preposition omitted (rule 25).

Maybe the strangest feature is the structure of PP. English PP is always P+NP. But here in Classical Chinese, by inversion and omission, the PP may have up to 4 forms, as shown in rule 22-25.

Table 4. The 4 rules from PP. The object of the preposition is in brackets, and [] indicate an omission.

No.	Rule	Explanation	Example
22	PP -> prep NP	The normal P+NP	儒*以(文)*乱法。
23	PP -> NP prep	Inverted: NP+P (介词宾语前置)	此*(所)以*乱也。
24	PP -> prepb	The object of the preposition is omitted	仲尼*以*([])为孝。 Omit [之]
25	PP -> NP	The preposition itself is omitted	谒之[]*(吏)* 。 omit [于]

Another feature that must be pointed out here is the cycle. In our rule-set, there are 2 rules (rule 11 and rule 17) forming a cycle:

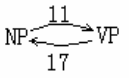


Fig. 4. A cycle in the rule-set. Rule 11: NP-> VP, Rule 17: VP-> NP.

It will ease our parsing because Classical Chinese is lexically and syntactically very ambiguous. An NP can act as a VP (a main verb), while a VP can act as a NP (subject or object). These two features are exemplified in figure 3. There are actually more cycles in the rule-set. Helpful as they are, the cycles bring great difficulty to the memory-based top-down parser. In practice, we develop a closure–based method to solve this problem, as shown in the following pseudo-code:

```
better_results_found=true;
while (better_results_found)
{
        better_results_found=false;
        memory_based_top_down_parse();
        // if better results found, the variable will be set true
}
```

4 Three-Level PCFG Model

4.1 Inspiration from Classical Chinese

Two-level PCFG parsing made a lot of errors, some of which reveal the intrinsic drawback of the model to us and inspired us of the novel three-level model.

There are 4 forms of PP (rule 22-25), and 2 forms of PP attachment (rule 9 and 10). Intuitively there will be 4*2=8 possible forms of combinations. But if we take a close look into the Classical Chinese language, we will find that for a pre-positioned PP, the PP's preposition is never omitted, that is, rule 9 never comes together with rule 25. And for a post-positioned PP, the PP is never inverted, that is, rule 10 never comes together with rule 23. In addition, rule 10 and rule 24 never comes together. So actually only 5 out of 8 forms of combinations exist.

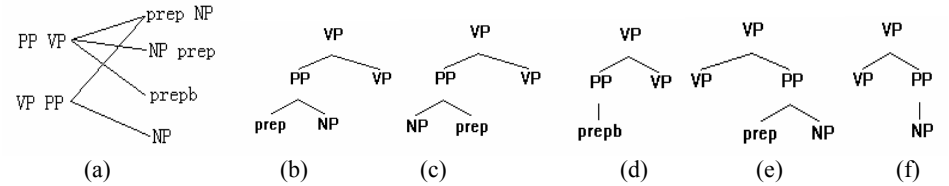


Fig. 5. 5 possible forms of PP attachment and expansions. (a) 5 combinations (b)-(f) corresponding parse trees

It is obvious here that traditional two-level CFG structure could not capture this information. It will guess for all the 8 combinations. So we need a more powerful model. It should be a three-level structure in order to describe the combinations. So we first define the *three-level CFG rule*, and we reserve the term *two-level rule* for the traditional CFG rule defined in Section 3.1.

4.2 Formal Definitions

Three-level Rule: A three-level rule *t* is an ordered pair of two two-level rules (R_1, R_2), where $R_1 : N_1 \to \alpha$, and $R_2 : N \to \beta$, such that $N \in \alpha$. R_1 and R_2 are called *base-rules* of *t*, and we denote $parentrule(t) = R_1$, $mainrule(t) = R_2$, $\rho(t) = LHS(R_1) = N_1$, $\mu(t) = LHS(R_2) = N$, and it is obvious that $\rho(t) = \rho(\mu(t))$. (See equation 1)

Three-level CFG: A three-level context-free grammar (three-level CFG) based on a CFG (V_N, V_T, S, R) is a quintuple (V_N, V_T, S, R, T), T is a set of three-level rules, such that $\forall t \in T$, $parentrule(t) \in R \wedge mainrule(t) \in R$

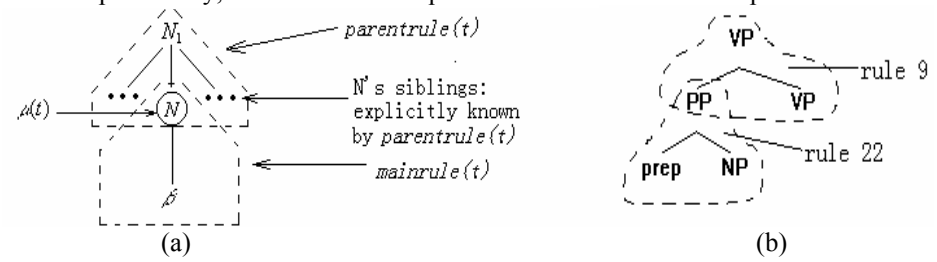
Three-level PCFG: A three-level probabilistic context-free grammar (three-level PCFG) is a six-tuple (V_N, V_T, S, R, T, P), where (V_N, V_T, S, R, T) is a three-level CFG, and $P : T \mapsto (0, 1]$ such that $\forall N \in V_N, \forall r \in R$,

$$\sum_{t:\, \mu(t)=N \wedge parentrule(t)=r} P(t) = 1 \qquad (2)$$

Where

$$P(t) = P(mainrule(t) \mid parentrule(t)) = \frac{f(mainrule(t),\ parentrule(t))}{f(parentrule(t))} \qquad (3)$$

Equation 3 is the basis of our three-level PCFG model, which denotes the probability of a three-level rule by a conditional probability, and it is also the equation we use to estimate the probabilities from the treebank.



(a) (b)

Fig. 6. (a) An illustration of the three-level rule and its base-rules. (b) An example: $P(t) = P(PP \to prep\ NP \mid VP \to PP\ VP)$

4.3 Top-down Parsing Algorithm for the Three-Level PCFG

1. The Algorithm

Our parsing algorithm is dynamic programming in nature. Based on the definitions of three-level rule in the previous subsection, we can develop a top-down algorithm which recursively passes the *parentrule* to the next level. Note that considering the *parentrule* implies knowing the parent non-terminal (as of [8]), as the *parentrule* is uniquely numbered.

We denote the probability $P(left, right, c, pr)$ to be the probability of the string $w_{left}, .., w_{right}$ to be of a non-terminal c under the condition that the parent rule of c is pr in the derivation. It is also the state-space of this algorithm.

Each time we select a rule r to expand the non-terminal c, using the probability of the three-level rule $P(r \mid pr)$. If r is a unary rule like $r \to A$, then

$$P(left, right, c, pr) = \max(P(left, right, c, pr), P(r \mid pr) \cdot P(left, right, A, r)) \qquad (4)$$

If r is a binary rule like $r \to AB$, then

$$P(left, right, c, pr) = \max(P(left, right, c, pr), \\ P(r \mid pr) \cdot P(left, mid, A, r) \cdot P(mid + 1, right, B, r)) \qquad (5)$$

For the implementation of the algorithm, we use memory-based recursive top-down parsing instead of bottom-up parsing because in this way the algorithm is more efficient by pruning those impossible constituents.

2. Complexity Analysis

We now analyze the time and space complexity of this memory-based top-down dynamic programming algorithm. Note that all the complexities we talked in this paper are the parsing complexities, not learning complexities.

Let M be the number of rules in the rule-set. Let n be the length of the sentence. Let H be the number of tags (non-terminals and terminals).

Space complexity: The state-space in this algorithm is provided by the probability $P(left, right, c, pr)$, so the space complexity is $O(n^2 \cdot H \cdot M)$.

Time complexity: The time complexity is the space complexity plus the nonterminal expansion complexity. Then the time complexity is $O(n^2 \cdot H \cdot M \cdot n \cdot M) = O(n^3 \cdot H \cdot M^2)$

4.4 Comparisons with Other Grammar Models

In this subsection we will focus interest on the grammar models and efficiencies only. We will leave accuracies to Section 5.

1. Comparison with pure two-level PCFG

It is very obvious that our three-level PCFG will significantly outperform the original two-level PCFG in parsing accuracies. The major reason for this improvement is the intrinsic "context-sensitivity" in our novel model. Theoretically, the new model is not a context-sensitive grammar, but still a pseudo context sensitive model. But by means of taking the parent rule into consideration, we may capture a lot of contextual information, as we explicitly know the siblings of a nonterminal when expanding it.

However, in terms of time and space complexity, we admit that the new model is a little bit more complex than the traditional one. For space complexity, the top-down parsing for two-level model is $O(n^2 \cdot H)$, compared with the new model's space complexity $O(n^2 \cdot H \cdot M)$. And for time complexity, it is $O(n^3 \cdot H \cdot M)$, compared with $O(n^3 \cdot H \cdot M^2)$ for the new model. We conclude here that the new model is just M times slower than the traditional model, and if we use some techniques to localize M to all non-terminals, it will be much smaller. And the accuracy gain is quite significant, as will be shown in Section 5. So it is worth applying the new model for the sacrifice of parsing speed.

2. Comparison with Charniak [8]'s pseudo context-sensitive model

Our model is a large improvement, compared with Charniak [8]'s model. We have stated in the introduction that only taking parent non-terminal into account could not discriminate different siblings, i.e. contextual information. The success of [8] may lies in the difference between Classical Chinese and English. For English, considering the parent is almost sufficient to discriminate siblings, while for Chinese, it is very far enough. We have seen in figure 5 that all the PPs are generated by a VP, but by different rewriting rules (rule 9 and 10), the situation is totally different. The model in [8] could not distinguish these different combinations. And from our corpus, we found that in Chinese, the situation that different rewriting rules from the same parent generating the same child is very common. So if used on Classical Chinese, Charniak [8]'s model will be just slightly better than the pure two-level PCFG, as is shown in Section 5. For complexity analysis, please refer to table 5. We admit that her model is a bit more efficient than ours.

3. Comparison with feature-based grammar

Generally feature-based grammar could be more accurate than probabilistic grammars. But it needs much more linguistic sophistications, i.e. grammatical features, resulting in a surprisingly large feature-based rule-set. And in Chinese, especially in Classical Chinese, such common features as AGRs do not exist. Actually Chinese is much more flexible in grammar than English, and Classical Chinese is even more flexible. So feature-based grammar is not so useful in Classical Chinese. Our three-level PCFG model, on the other hand, incorporates some of the features into grammar model. The example shown in figure 5 is, to some extent, a grammatical feature.

4. Comparison with n-gram model

Our three-level model is also inspired by the trigram model. Although the n-gram model is commonly used in Part-of-Speech tagging, we here incorporate its idea into parsing. We think the two-level model is very much similar to the bigram model, and the three-level model to the trigram model. Both three-level and trigram model depend the selection of current node on the previous two, rather than one, nodes. Both models are efficient, while requiring a larger space complexity for storing the information learned from the training set. The major difference between three-level/trigram and two-level/bigram, as far as we think, is the amount of information

learned from the same corpus. Actually the three-level/trigram model *mined* more *implicit* information (three-level probabilities), in addition to the explicit information also learned by two-level/bigram model.

5. Comparison with DOP or STSG

The DOP framework proposed in [10] and [11] and the instantiation by R. Bod [11] provide a brand new model for accurate parsing in a specific domain. The grammar model used by Bod is Stochastic Tree Substitution Grammar (STSG), where all subtrees are considered units of substitution. Intuitively the read might think that STSG is an expansion of our three-level PCFG, as STSG considers subtrees of all levels. But a deeper thought will result in the idea that STSG is still bigram-like in nature. Just consider the following definition of STSG probability [11]:

STSG: An STSG is a quintuple (V_N, V_T, S, C, PT) based on the CFG (V_N, V_T, S, R), where C is a set of subtrees, such that the rule-set R contains all and only those rules involved in the subtrees of C. PT assigns to every $t \in C$ a value $0 < PT(t) \le 1$ such that $\forall N \in V_N : \sum_{t \in C, root(t)=N} PT(t) = 1$

We have seen from the above definition that STSG only considers the root node, and ignores its parent and parent rule (parent subtree, in this sense), so it is not beyond bigram in nature, and different from ours. In addition, the major drawback of STSG is its inefficiency: it has exponential time complexity.

A conclusion of these models is shown in the following table and figures.

Table 5. Complexity Comparisons (parsing complexities)

Model	Pure two-level PCFG	Charniak [8]	DOP or STSG	Our three-level PCFG
Space Complexity	$O(n^2 \cdot H)$	$O(n^2 \cdot H^2)$	Exponential	$O(n^2 \cdot H \cdot M)$
Time Complexity	$O(n^3 \cdot H \cdot M)$	$O(n^3 \cdot H^2 \cdot M)$	Exponential	$O(n^3 \cdot H \cdot M^2)$

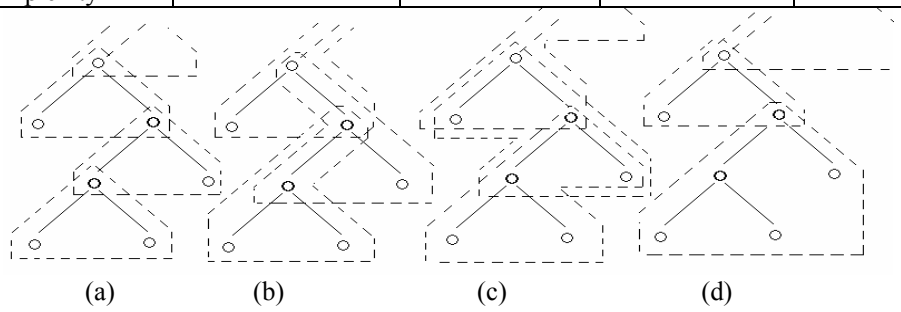


Fig. 7. Tree illustration of the 4 grammar models. (a) two-level PCFG (b) Charniak [8] (c) our three-level PCFG (d) STSG

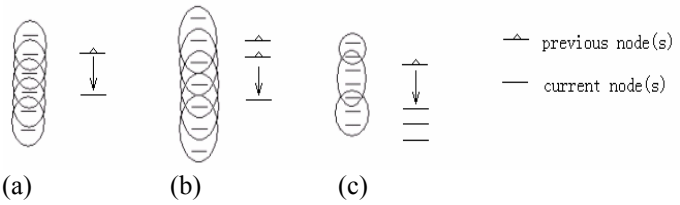


Fig. 8. Linear Illustration of the grammar models. (a) bigram model and two-level PCFG, (b) trigram model, our three-level model, Charniak [8], and (c) STSG. The most important difference lies in the intersection of two consecutive circles, i.e. phases in parsing: for (a) and (c), the intersection contains one node; for (b), the intersection contains 2 nodes. And to this end STSG is not beyond two-level PCFG.

5 Experiments and Results

In our preliminary experiments, we constructed a treebank of 5000 manually parsed sentences, in which 500 sentences are selected as the test set using the cross-validation scheme, while the others as the learning set. The majority of these sentences are extracted from classics of pre-Tsin Classical Chinese such as *韩非子* and *荀子* because in these texts there are fewer proper nouns and difficult words. It must be pointed out here that

compared from other languages, Classical Chinese sentences are so short that the average length is only about 4-6 words long.

We applied four parsing schemes to these texts: (1) the pure two-level PCFG, (2) the complete model of two-level PCFG where predefined preferences are added to correct some typical errors of the pure PCFG, (3) the model proposed by Charniak in [8] based on the complete two-level model, (4) our innovative three-level PCFG model where no predefined features are incorporated.

5.1 Accuracies

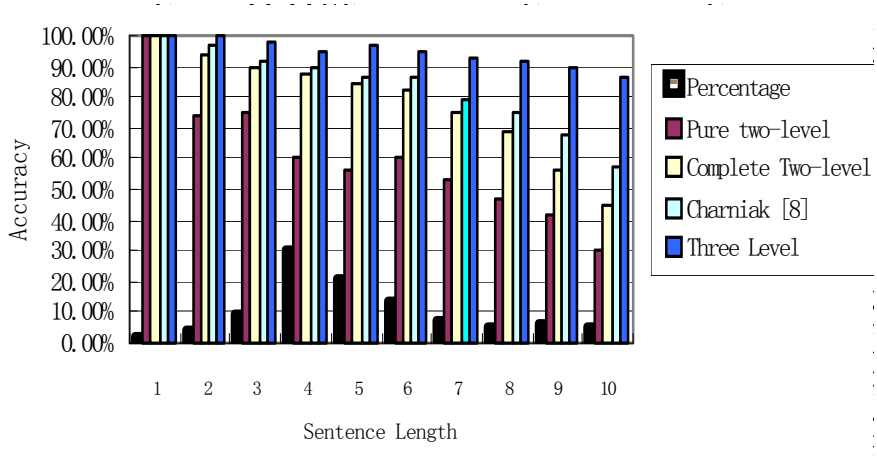


Fig. 9. Sentence length distributions and accuracies of different parsing schemes mentioned above

Figure 9 shows the distribution of sentences and parsing accuracies by the four different models. For distribution, we can see that those 4-word, 5-word, and 6-word sentences constitute for the majority of the corpus. For accuracies, the pure two-level PCFG is certainly the worst for an overall accuracy of 58.6%. The complete two-level model does correct many errors and improve the overall accuracy to be 81.0%. Charniak's pseudo context-sensitive model [8] maybe effective for English, but here in Classical Chinese we found not so successful. It only improves the complete model a little bit, to 84.8%. Our three-level model is surely the best for an overall accuracy of 94.7% while no predefined features are embedded. On the other hand, when the sentence length is enlarging, the differences among these 4 models are becoming more obvious. For sentences of 10 words long, the accuracy of pure two-level PCFG is only 30.0%, while our three-level model still gets a promising result of 86.2%.

5.2 Learning Curves

To test the learning curves, we let the training set enlarging from 1000 sentences to 4500 sentences and record the accuracies of every 500 sentences for each parsing scheme. From the result shown below, we found that the learning curve of three-level model is very similar to the curve of trigram model.

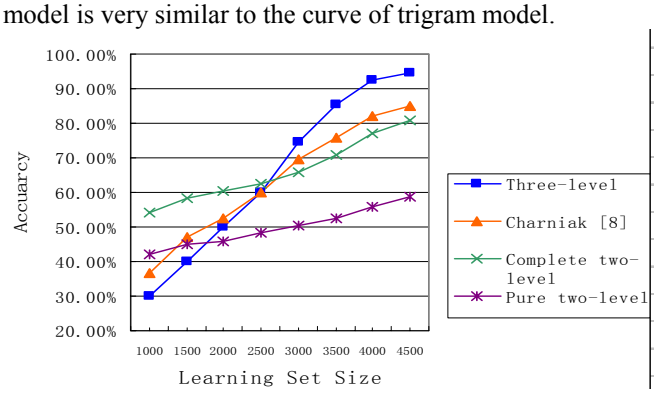


Fig. 10. Learning Curves of the four different parsing schemes.

From the above figure, we can divide these four models into 2 categories: The pure two-level PCFG and complete two-level model are *two-level models*; while Charniak [8]'s and ours are *three-level models*. The curve

of the complete two-level is almost parallel to the pure two-level model, since they are very similar in the nature. At the beginning, two-level models substantially outweigh the three-level models, mostly because the *sparse-data-problem* occurs, as it does in the trigram model for POS tagging. But as the learning set grows larger, three-level models significantly outperform their two-level counterparts, because then the corpus is able to provide enough contextual information. This comparison is very similar to the one often conducted between bigram and trigram in POS tagging. In addition, between the two three-level models, the reader might found that our model is more *trigram-like* as shown by its curve. Actually based on the discussion in previous sections, one will conclude that the model in [8] is just a pseudo three-level one, and requires much less contextual information than ours.

6 Conclusion

In this paper, we have proposed and analyzed a trigram-inspired three level PCFG model and its successful application in parsing Classical Chinese texts.

To improve the accuracy of the traditional PCFG model, we devised a brand new three-level PCFG model that is to some extent quite "context-sensitive". Charniak [8] proposed a "pseudo context-sensitive" model that considers parent nonterminal when expanding a nonterminal. It does provide some contextual information but we have shown that it ignores the siblings' information and is especially unsuccessful in parsing non inflexion-based languages such as Chinese. Now in our model, the expansion of a non-terminal *N* depends not only on its parent, but also the rule used by the parent to generate *N*. In this way we can gain much more context-sensitivity than the model of [8]. For its applications, we found it especially successful in parsing Classical Chinese texts. In our preliminary experiment, we got a promising result of 94.7% for overall accuracy, which is substantially higher than other models.

Another contribution of this paper lies in the computer processing of Classical Chinese. We continue to use the tagset and corpus of previous work by L. Huang et al. [4] into this study. A PCFG model is presented where we restrict the rules into binary/unary rules, but no so strict as CNF. A rule-set is designed for Classical Chinese which have been shown to be of a broad-coverage in our corpus. Some special features in the rule set are discussed in this paper. The analyzing of the typical errors made by the two-level PCFG also inspired us of the three-level model.

As a whole, this three-level model is, in almost all aspects, very similar to the trigram model in POS tagging. They share the same mining technique of corpus information, the dynamic programming algorithm, the high accuracy, and the learning curves. So we say that this three-level model is actually a trigram-inspired model.

7 References

[1] J. Allen: Natural Language Understanding, The Benjamin/Cummings Publishing Company, Inc. 1995

[2] A. Viterbi: Error bounds for convolution codes and an asymptotically optimal decoding algorithm. IEEE Trans. on Information Theory 13:260-269. 1967

[3] Y. Yao, K. Lua: A Probabilistic Context-Free Grammar Parser for Chinese, Computer Processing of Oriental Languages, Vol. 11, No. 4, 1998, pp. 393-407

[4] L. Huang, Y. Peng, H. Wang: Part-of-Speech Tagging for Old Chinese, to appear in the 5th International Conference on Text, Speech, and Dialog (TSD), Brno, 2002, manuscript available at blhuang@online.sh.cn

[5] D. Klein, and C. Manning: Natural Language Grammar Induction using a Constituent-Context Model, Proceedings of Neural Information Processing Systems (NIPS 2001), Vancouver, 2001.

[6] Y. Yao, K. Lua: Mutual Information and Trigram Based Merging for Grammar Rule Induction and Sentence Parsing, Computer Processing of Oriental Languages, Vol. 11, No. 4, 1998, pp. 393-407

[7] M. Johnson: Joint and conditional estimation of tagging and parsing models, Proceedings of International computational linguistics conference (ACL'01), Toulouse, 2001

[8] E. Charniak: Context-Sensitive Statistics for Improved Grammatical Language Models, In Proceedings of AAAI-94, pages 728-733, 1994.

[9] E. Charniak: Taggers for Parsers, Artificial Intelligence, Vol. 85, No. 1-2, pp. 45-47, 1996.

[10] R. Scha: Language Theory and Language Technology; Competence and Performance (in Dutch). In Q.A.M. de Kort and G.L.J. Leerdam, editors. Computertoepassingen in de Neelandistiek. 1990

[11] R. Bod: Enriching Linguistics with Statistics; Performance models of Natural Language. Ph.D. thesis, University of Amsterdam, 1995.