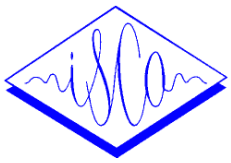


3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

DYNAMIC PROBABILISTIC GRAMMAR FOR SPOKEN LANGUAGE DISAMBIGUATION

Takeshi Kawabata

NTT Basic Research Laboratories
3-1 Morinosato-Wakamiya, Atsugi-shi, 243-01 JAPAN

ABSTRACT

The dynamic probabilistic grammar (DPG) is a context-free grammar whose rule probabilities are dynamically controlled by a hidden Markov model (HMM). This HMM receives a rule number as an input symbol, and outputs the probability of the rule number sequence. The DPG parser generates plural rule sequences representing different syntactic structures. The parser calculates the probability for each rule sequence, and selects the rule sequence (i.e. syntactic structure) which achieves maximum probability. This disambiguation mechanism is also effective for grammar-based speech recognition systems. The number of candidate words (called perplexity) can be reduced effectively using this mechanism. The DPG provides a stochastic framework for CFG-class spoken language processing.

1. INTRODUCTION

Syntactic disambiguation is an important issue in grammatical natural language processing. For example, the context-free grammars (CFG) which cover natural language variations are often ambiguous in the sense that one sentence (a word sequence) can be generated by more than one grammar rule sequence. Syntactic coverage and disambiguity is not compatible for simple CFGs. The augmented CFG with semantic restrictions is one solution of this problem, and several unification-based approaches have been developed [1]. These approaches are effective, but require a lot of effort to use in the design of complex semantic/syntactic feature structures. Another solution is a probabilistic CFG whose grammar rules have production probabilities [2]. The probability of a given sentence is calculated via each grammar rule sequence. The rule sequence which achieves maximum probability determines the syntactic structure for the given sentence.

This paper introduces a technology that can be used to reduce CFG ambiguities. The dynamic probabilistic grammar (DPG) is a context-free grammar whose rule probabilities are dynamically controlled by a hidden Markov model (HMM). This HMM receives a rule number as an input symbol, and outputs the probability of the rule number sequence. The DPG parser generates plural rule sequences representing different syntactic structures for an ambiguous incoming sentence. The parser calculates the HMM probability for each rule sequence, and selects the rule sequence (i.e. syntactic structure) that achieves maximum probability. The rule HMM is trained in advance, using the text corpus with hand-labeled syntactic trees.

This disambiguation mechanism is also effective for use in grammar-based speech recognition systems. Using this mechanism, the number of candidate words (called perplexity) can be reduced. Traditional HMM-based word sequence models can only treat regular grammar(RG)-class languages. The DPG provides a stochastic framework for treating CFG-class languages.

2. DYNAMIC PROBABILISTIC GRAMMAR

2.1 Basic Concept

Figure 1 shows the schematic diagram of the dynamic probabilistic grammar (DPG). The CFG parser analyzes an input word sequence and generates grammar-rule sequences. Each rule sequence uniquely corresponds to a syntactic structure. These rule sequences drive the hidden Markov model and the current rule context is estimated as a HMM state distribution. According to this HMM state distribution, the CFG rule probabilities are dynamically changed . The parser calculates the total probability for each rule sequence and selects the rule sequence (i.e. syntactic structure) which

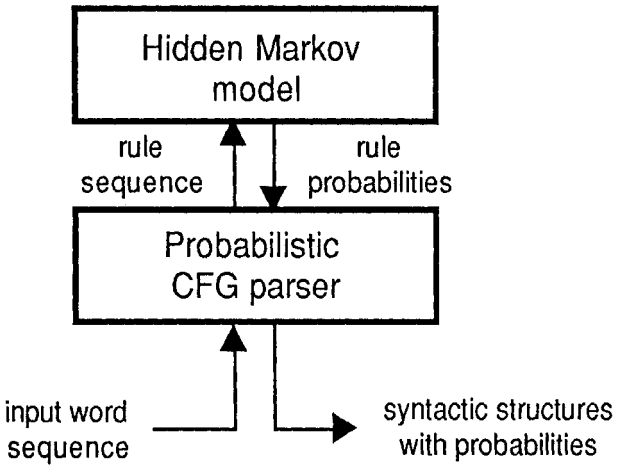


Fig. 1 Schematic diagram of dynamic probabilistic grammar

achieves maximum probability.

Figure 2 shows how the DPG system works. A context-free grammar is prepared for analyzing input word sequences. A simple toy grammar is shown in Fig. 2. The parser analyzes the Japanese phrase "bunka ga (The culture is ...)" using this grammar and generates the rule number sequence "1, 2, 3, 6".

The rule HMM is the ergodic hidden Markov model driven by grammar-rule sequences. The initial state-probability distribution of this HMM is set to be uniform. This means that the rule context is unknown at the beginning of the parsing process. By receiving rule number tokens, the HMM state-probability distribution is changed according to the model parameters. Thus, the rule contexts reflect the state distribution. Each HMM state has a probability table for grammar rules. Taking the state distribution into account, this table is merged and each rule probability for the next parsing step is calculated. The probability of a syntactic structure is the product of the rule probabilities for the corresponding rule sequence.

2.2 Rule Hidden Markov Model

The rule HMM is a stochastic state transition model with an ergodic structure. An example of a two-state ergodic HMM is shown in Fig. 3. Here, all arcs toward the same states have the same output probability table (tied arcs) used to reduce the degree of freedom.

The rule HMM is trained using the text corpus with hand-labeled syntactic trees. First, we made a simple context-free grammar that covers all sentences in this corpus. Lexical rules were generated by collecting different words according to their part of speech (POS) labels. Intra-phrase rules were generated by checking the connectivity of these labels. The grammar also contains simple inter-phase rules. Concatenation of an arbitrary number of phrases results in either a sentence or a partial sentence. Figure 4 shows the skeleton of this grammar.

The word sequences of this corpus are analyzed by this simple CFG with POS labels and transformed into rule number sequences. The rule HMM is trained on these rule sequences by using the forward-backward algorithm [3].

2.3 Dynamic Probabilistic Grammar

Each grammar rule in a static probabilistic grammar has a fixed production probability. The sum of rule production probabilities for the rules having the same left-hand symbol is normalized to 1.0. The rule production probabilities for the DPG change dynamically according to the rule HMM state distribution.

The state probability $\alpha(i,t)$ (of state i at frame t) is calculated by the following procedure. First, at t=0 set

$$\alpha(i,0) = 1/(number\ of\ states) \qquad (1)$$

for each state. By receiving the rule number k, the HMM

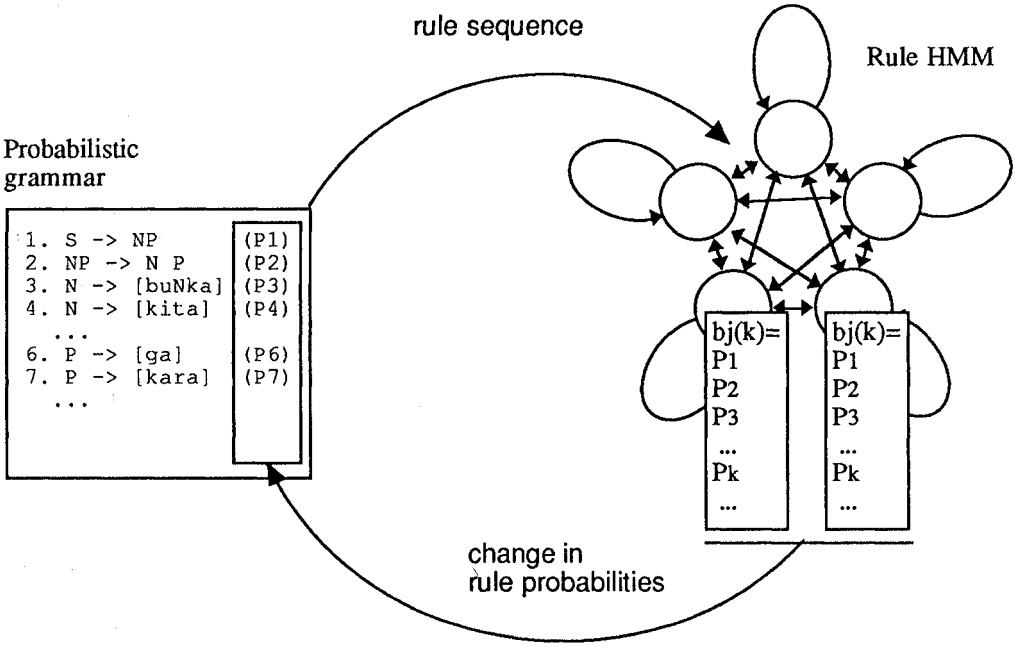


Fig. 2 How the dynamic probabilistic grammar works

state-probability distribution changes according to transition probabilities and output probabilities.

$$\alpha(i,t) = \sum_j \alpha(j,t-1) a_{ji} b_{\cdot i}(k) \qquad (2)$$

where

$\alpha(j,t)$: forward probability (at state j at frame t)

a_{ji} : transition probability (from state j to i)

$b_{\cdot i}(k)$: output probability (to state i for rule k)

This state distribution reflects the rule context. The production probability of rule k for the next parsing step i is calculated as

$$P(k,t) = \frac{\sum_i \sum_j \alpha(j,t-1)\, a_{ji} b_{\cdot i}(k)}{\sum_l \sum_i \sum_j \alpha(j,t-1)\, a_{ji} b_{\cdot i}(l)} \qquad (3)$$

where Σ_l means the sum through the rules which have the same left-hand symbol. This normalization guarantees the consistency of the DPG.

The probabilistic grammar G is consistent [2] if

$$\sum_{x \in L(G)} p(x) = 1 \qquad (4)$$

where x is a sentence and $L(G)$ is the language generated by the grammar G. Let the time axis t be the number of rules used for generating a symbol. Let V be the set of non-terminal symbols of grammar G, and let W be the set of terminal symbols. The sum of the probabilities for all the sentences derived from symbol $X\ (\in V \cap W)$ is calculated as

$$P(X) = \sum_l p_l(R_1, \cdots, R_{t-1}) \prod_i P(Y_{li}) \qquad (5)$$

where Y_{li} is the i-th right-hand symbol of rule l, Π_i means the product of the right-hand symbols of rule l, and Σ_l means the sum of the rules whose left-hand symbol is X. Let $p_l(R_1, \cdots, R_{t-1})$ be the dynamic production probability of rule l according to the rule history. For any terminal symbol X, let

$$P(X) = 1 \quad (for\ X \in W). \qquad (6)$$

Assume

$$\sum_l p_l(R_1, \cdots, R_{t-1}) = 1. \qquad (7)$$

From Equation (5), (6) and (7), we obtain

$$P(S) = 1 \quad (S : start\ symbol) \qquad (8)$$

where S is the start symbol for grammar G. Equation (8) has the same meaning as Equation (4). Thus, Equation (6) and (7) provide a sufficient condition for the consistency of the dynamic probabilistic grammar.

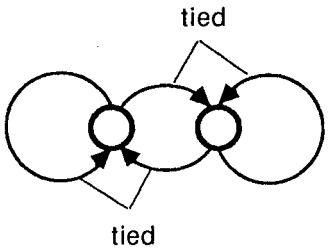


Fig. 3 Tied arcs of rule HMM

3. EVALUATION

The text corpus for training and evaluation was collected through a dialog simulation of a secretarial service at an international conference. Two persons communicate through video display terminals. One plays the role of a secretary of an international conference, and the other plays an applicant. Dialogs including over 23,000 partial sentences were collected. A partial sentence, in this paper, is defined as a unit delimited by punctuation marks.

```
;; inter-phrase rules
  sentence --> ps$ .
  ps$ --> ps .
  ps$ --> ps , ps$ .
;; intra-phrase rules
  ps --> np$ .
  ps --> vp$ .
  ...
  np$ --> n .
  np$ --> n , p$ .
  np$ --> n , suf  .
  ...
p$ --> p .
p$ --> p , p$ .
...
vp$ --> v .
vp$ --> v , aux$ .
vp$ --> v , suf  .
...
aux$ --> aux .
aux$ --> aux , aux$ .
aux$ --> aux , suf  .
...
;; lexical rules
  n --> [bunka] .
  n --> [kita] .
  ...
  p --> [ga] .
  p --> [kara] .
  ...
  v --> [tsutawaru] .
  v --> [tsutawaQ] .
  ...
  aux --> [ta] .
  ...
```

Fig. 4 Simple context-free grammar for spoken language

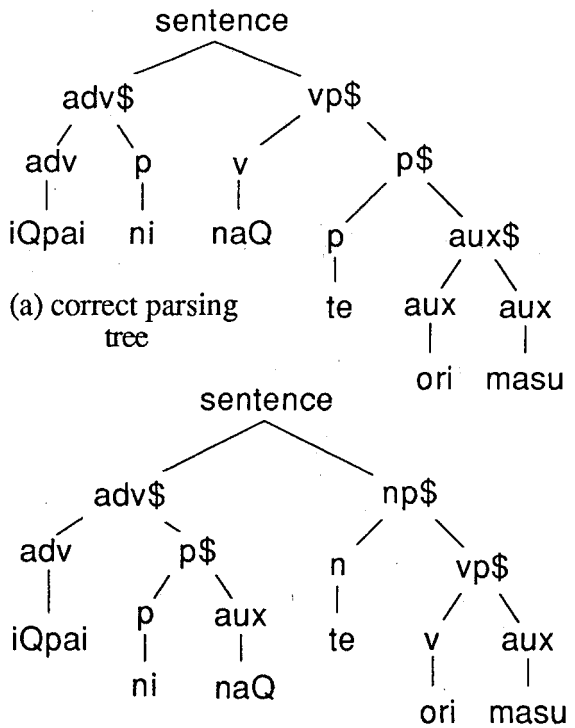
The DPG is trained with 20,000 partial sentences. All words in the training sentences are manually POS-labeled in advance. The word sequences with labels are analyzed by the simple CFG and transformed into rule number sequences. The rule HMM is trained on these rule sequences by using the forward-backward algorithm [3].

Figure 5 shows examples of syntactic trees for the same word sequence. When POS labels are applied, the grammar can generate a correct unique tree (Fig. 5(a)). The ambiguous CFG without POS labels, however, generates 291 different trees for this word sequence. Even in this case, the DPG (without labels) can choose the correct parsing tree.

Quantitative evaluation of the DPG method is carried out using 3,000 partial sentences. These test sentences are different from the training data. The ambiguity of the grammar is measured with the logarithm scale as

$$H = -\frac{1}{N_T}\sum_{s\in T}\log_2 p(s), \qquad (9)$$

where s is a sentence in the test set T, and N_T is the number of sentences in T. A larger H indicates that the grammar is more ambiguous. To compare the simple CFG and DPG, the difference of log sentence probabilities $\log_2 p_{CFG}(s) - \log_2 p_{DPG}(s)$ is plotted in Figure 6 for each sentence. The DPG effectively reduces grammar ambiguities. This figure also shows that the DPG is more effective for longer sentences.



(a) correct parsing tree

(b) an example of an inadequate tree

Fig. 5 Variation of syntactic trees for the same word sequence

The disambiguation mechanism of DPG is also effective for grammar-based speech recognition systems. For this purpose, the number of candidate words (perplexity) is important. The perplexity is defined as

$$F = \exp_2\left(-\frac{1}{N_T}\sum_{s\in T}\frac{1}{N_s}\log_2 p(s)\right), \qquad (10)$$

where N_s is the number of words in the sentence. The perplexity reduction ratio between the simple CFG and DPG was about 1/4.

CONCLUSION

The dynamic probabilistic grammar (DPG) for spoken language disambiguation has been described. By combining stochastic and grammar-based approaches, we constructed an effective language model for real-world speech. Evaluation experiments were carried out using a large dialog-text corpus. The DPG effectively reduced grammar ambiguities for sentences of various lengths. Perplexity reduction experiments were also carried out using the same corpus. The DPG reduced the perplexities by about 1/4

REFERENCES

[1] Sells, P. : "Lectures on Contemporary Syntactic Theories," CSLI Stanford Univ. (1985)

[2] Fu, K. S.: "Stochastic Languages for Picture Analysis," Computer Graphics and Image Processing, 2, pp. 433-453 (1973).

[3] Levinson, S. E., Rabiner, L. R., and Sondhi, M. M.: "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech recognition," BSTJ, Vol. 62, No. 4, pp. 1035-1074 (1983)

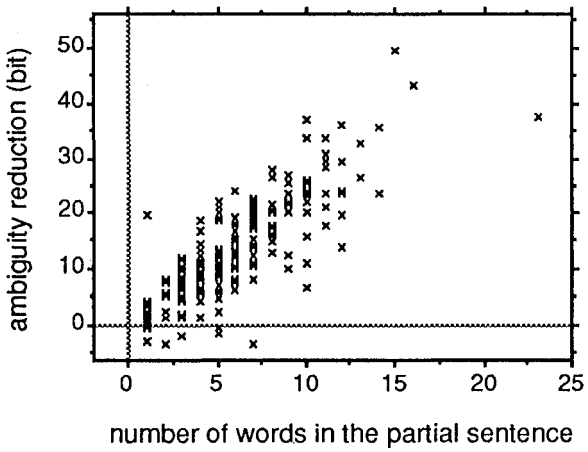


Fig. 6 Reduction of syntactic ambiguity