Language Modeling by Stochastic Dependency Grammar for Japanese Speech Recognition

Akinori Ito†, *Chiori Hori*‡, *Masaharu Katoh*† *and Masaki Kohda*†

†Faculty of Engineering, Yamagata University ‡Tokyo Institute of Technology

ABSTRACT

This paper describes a language modeling technique using a kind of stochastic context free grammar (stochastic dependency grammar, SDG). In this work, two improvements are done upon the general CFG based SCFG model. The first improvement is to use a restricted grammar instead of general CFG. The dependency grammar used here is a restricted CFG that expresses modification between two words or phrases. The derivation probabilities are estimated by inside-outside algorithm. The computational complexity of the estimation is reduced from $O(N^3L^3)$ to $O(N^2L^3)$, where N and L means the number of nonterminals and length of a sentence respectively. Second, word grouping is introduced for further reduction of the estimation time. The basic idea is that regular grammar is applied within a group and CFG is used to express intergroup relationship. To achieve the idea, a new algorithm is introduced. When a group have two words in average, the learning time becomes about one-eighth. Two experiments were carried out to investigate the performance of the proposed model. In the first experiment, various kinds of SCFGs were compared using perplexity. From the result, it was found that the proposed model have much lower PP than the original model. As for the training speed, restricted grammar made training process twenty times faster, and the word grouping made it eight times faster. In the second experiment, the proposed model was used as a language model of LVCSR. The result showed that the proposed model was as good as bigram and trigram, and that the combination of trigram and the proposed model achieved further improvement of WER.

1. Introduction

Language model is an indispensable component for large vocabulary continuous speech recognition (LVCSR). N-gram based language models are most popular for that purpose. As n-gram models, especially bigram and trigram models, are simple and powerful, almost all LVCSR systems exploit bigram and/or trigram models. However, there have been a criticism against n-gram models that it can't reflect long-range dependency because n-gram models use information of only a few preceding words to predict probability of the current word.

Introduction of context free grammar (CFG) is the most straightforward way to exploit long-range dependency. A CFG is extended to a stochastic context free grammar (SCFG) to give a probability of a sentence. An SCFG G is expressed as

$$G = (\mathcal{N}, \mathcal{T}, \mathcal{P}, S) \tag{1}$$

where $\mathcal{N}, \mathcal{T}, \mathcal{P}$ are set of nonterminal symbols, terminal symbols and production rules with derivation probabilities respectively, and S stands for the starting symbol. If we have many parse trees as training data, an SCFG can be estimated using ML estimate. However, it is not easy to obtain such parse trees. The inside-outside (IO) algorithm[1, 2] is an EM-based algorithm that makes it possible to estimate an SCFG from any texts without parse tree.

The biggest problem of IO algorithm is that it is computationally very expensive. Its computational complexity is $O(N^3L^3)$, where N is number of nonterminal symbols and L is length of given sentence. When number of nonterminals gets larger, learning process becomes slower very rapidly. If we use a grammar for NLP, number of nonterminals should be more than several hundred, which is impossible to estimate probabilities using IO algorithm.

In this work, two kind of improvements are done upon traditional IO based SCFG. The first point is to use restricted grammar to reduce computational complexity of IO algorithm. The second is to group phrase before applying IO algorithm. As a Japanese phrase contains approximately two words in average, the computation time for training gets about one-eighth of original IO algorithm.

2. Stochastic Dependency Grammar

The IO algorithm requires computational complexity of $O(N^3)$ where N is number of nonterminals, because number of parameter $P(\alpha \to \beta\gamma)$ is in proportion to N^3. If number of rules can be reduced, the computational complexity can be reduced too. In this work, number of rules is reduced to N^2 by considering dependency structure of Japanese language.

Japanese language has a feature that a modifier precedes to the head. For example,

akai	*hana*	([a] red flower)
red	flower	

neko	*ga*	*iru*	([there] is [a] cat)
cat	case-SUB	be	

In the first example, the word *akai* modifies the word *hana*, which is the head of this noun phrase. In the second exam-

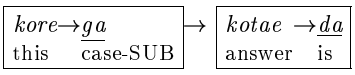

Figure 1: Phrases and their dependency in Japanese

ple, the word *neko* modifies the postposition *ga* which is a case marker of subject, and the phrase *neko ga* modifies the verb *iru*.

Considering this feature, a rule $\alpha \to \beta\gamma$ in Japanese language can be interpreted as "an element β modifies γ, and the whole string can be regarded as α." Now we introduce an assumption that the grammatical category of head and entire string is similar. Then the rule can be approximated as

$$\alpha \to \beta\alpha \tag{2}$$

which can be interpreted as "an element β modifies α, and the whole string can be regarded as α." This kind of modification is called *kakari-uke* in Japanese linguistics terminology. With this approximation, a rule contains up to two kind of nonterminals, which means that the computational complexity of training algorithm can be reduced to $O(N^2)$. We call this kind of grammar *stochastic dependency grammar*.

Many languages, including English, have both left-to-right and right-to-left modification. In this case, a grammar has to have rules that correspond to two kind of modifications:

$$\alpha \to \beta\alpha \tag{3}$$
$$\alpha \to \alpha\beta \tag{4}$$

With these rules, the complexity is still $O(N^2)$.

3. Consideration of phrase boundary

Another problem of IO algorithm is that the training time is in proportion to $O(L^3)$. To reduce the training time, further restriction is done upon the training algorithm and the grammar. The basic idea is that a sentence is divided into phrase-like unit (called *bunsetsu*) and inside and outside probabilities are calculated for *bunsetsu* sequence. In the following discussion, we denote this phrase-like unit by 'phrase.' As Japanese phrase contains approximately two words in average, the training time is expected to be reduced to about one-eighth.

Japanese phrase is defined as a content word followed by zero or more function words. Figure 1 shows an example of the phrases and their dependency. The example sentence is "*kore ga kotae da*" (this is [the] answer). In the sentence, "*kore ga*" and "*kotae da*" are grouped into phrases. The underlined words (*ga* and *da*) are function words. According to Japanese modification rule, a content word modifies the following function words and they forms one phrase. As a content word always starts a new phrase, it is very easy to divide a sentence into a phrase sequence.

To realize inter-phrase (or *inter-bunsetsu*) SCFG, we have to consider *intra-phrase forward probability* h and *intra-phrase backward probability* r, as well as further modification of SCFG rules.

To utilize inter-phrase dependency, the following three kinds of rules have to be used instead of Chomsky normal form:

$$\alpha \to \beta\alpha \quad \alpha,\beta \in \mathcal{N} \tag{5}$$
$$\alpha \to w_c \quad \alpha \in \mathcal{N},\ w_c \in \mathcal{T}_c \tag{6}$$
$$\alpha \to \beta w_f \quad \alpha,\beta \in \mathcal{N},\ w_f \in \mathcal{T}_f \tag{7}$$

where $\mathcal{N}$ is a set of nonterminals, $\mathcal{T}_c$ is a set of all content words and $\mathcal{T}_f$ is a set of all function words. The first and the second rules are almost same to that of Chomsky normal form, and the third rule introduces *intra-phrase* dependency in Japanese.

Let us define the following notations:

M	Number of phrases in a sentence
K_m	Number of function words in the m-th phrase ($K_m \geq 0$)
w_{mc}	The content word of m-th phrase
$w_{mf,i}$	The i-th function word of mth phrase
P_m	m-th phrase ($= w_{mc}w_{mf,1}\dots w_{mf,K_m}$)
$a(\beta\|\alpha)$	Production probability of the rule $\alpha \to \beta\alpha$
$b(w_c\|\alpha)$	Production probability of the rule $\alpha \to w_c$
$c(\beta w_f\|\alpha)$	Production probability of the rule $\alpha \to \beta w_f$
$h(m,i,\alpha)$	Intra-phrase forward probability of m-th phrase
$r(m,i,\alpha)$	Intra-phrase backward probability of m-th phrase
$e(m,n,\alpha)$	Inter-phrase inside probability
$f(m,n,\alpha)$	Inter-phrase outside probability

$$g(m,n,\alpha,\beta) = \sum_{l=m}^{n-1} a(\beta|\alpha)e(m,l,\beta)e(l+1,n,\alpha)$$

All probabilities are calculated as follows:

Intra-phrase forward probability

$$\begin{aligned} h(m,i,\alpha) &= P(\alpha \to w_{mc}w_{mf,1}\dots w_{mf,i}) \\ &= \begin{cases} b(w_{mc}|\alpha) & \text{if } i=0 \\ \sum_\beta h(m,i-1,\beta)c(\beta w_{mf,i}|\alpha) & \\ & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

Inter-phrase inside probability

$$\begin{aligned} e(m,n,\alpha) &= P(\alpha \to P_m \dots P_n) \\ &= \begin{cases} h(m,K_m,\alpha) & \text{if } m=n \\ \sum_\beta g(m,n,\alpha,\beta) & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

Inter-phrase outside probability

$$f(m,n|\alpha) = P(S \to P_1 \dots P_{m-1}\alpha P_{n+1}\dots P_M) \tag{10}$$

$$f(1,M,\alpha) = \begin{cases} 1 & \text{if } \alpha = S \\ 0 & \text{if } \alpha \neq S \end{cases}$$

$$\begin{aligned} f(m,n,\alpha) = &\sum_{l=1}^{m-1}\sum_\beta a(\beta|\alpha)e(l,m-1,\beta)f(l,n,\alpha) + \\ &\sum_{l=n+1}^{M}\sum_\beta a(\alpha|\beta)e(n+1,l,\beta)f(m,l,\beta) \\ &\qquad\qquad \text{otherwise} \end{aligned}$$

Table 1: Conditions of experiment 1

Number of nonterminals	20	
Vocabulary size	3032 (Number of distinct words occurred more than twice in the corpus)	
Corpus	EDR corpus (Japanese corpus from newspapers and magazines)	
	Training text	Evaluation text
# sentence	2000	100
# word	53910	2782
UNK ratio	10.3%	22.0%

Table 2: Compared SCFGs

		SCFG		
		word based		phrase based
restricted grammar not used	name	SCFG		P-SCFG
	rule type	$\alpha \to \beta\gamma$ $\alpha \to w$		$\alpha \to \beta\gamma$ $\alpha \to w$ $\alpha \to \beta w$
	complexity	$O(N^3L^3)$		$O(N^3M^3)$
restricted grammar used	name	K-SCFG	K-SCFG2	PK-SCFG
	rule type	$\alpha \to \beta\alpha$ $\alpha \to w$	$\alpha \to \beta\alpha$ $\alpha \to w$ $\alpha \to \beta w$	$\alpha \to \beta\alpha$ $\alpha \to w$ $\alpha \to \beta w$
	complexity	$O(N^2L^3)$		$O(N^2M^3)$

Intra-phrase backward probability

$$r(m,i,\alpha) = P(S \to P_1 \dots P_{m-1}\alpha \; w_{mf,i+1} \dots w_{mf,K_m} P_{m+1} \dots P_M) \quad (11)$$
$$= \begin{cases} f(m,m,\alpha) & \text{if } i = K_m \\ \sum_\beta c(\alpha w_{mf,i+1}|\beta) r(m,i+1,\beta) & \text{otherwise} \end{cases}$$

Using these probabilities, each parameter can be reestimated as follows:

$$a'(\beta|\alpha) = \frac{\sum_{m=1}^{M-1} \sum_{n=m+1}^{M} g(m,n,\alpha,\beta) f(m,n,\alpha)}{\sum_\beta \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} g(m,n,\alpha,\beta) f(m,n,\alpha)} \quad (12)$$

$$b'(w|\alpha) = \frac{\sum_{m:w_{mc}=w} b(w|\alpha) r(m,0,\alpha)}{\sum_{m=1}^{M} b(w_{mc}|\alpha) r(m,0,\alpha)} \quad (13)$$

$$c'(\beta w|\alpha) = \frac{\sum_{m=1}^{M} \sum_{i:w_{mf,i}=w} h(m,i-1,\beta) c(\beta w|\alpha) r(m,i,\alpha)}{\sum_{m=1}^{M} \sum_{i=1}^{K_m} h(m,i,\alpha) r(m,i,\alpha)} \quad (14)$$

4. Experiments

Two experiments were carried out to investigate the performance of the proposed model. In the first experiment, various kinds of SCFGs were compared using perplexity. Table 1 shows the conditions of the experiment.

In this experiment, five kinds of SCFGs were compared each other. Table 2 shows the specification of each models. The SCFG model is an original SCFG without any improvements. P-SCFG model refers phrase boundary, but the grammar is not restricted. K-SCFG uses the restricted grammar rules but it doesn't consider phrase boundary. K-SCFG2 uses phrase-conscious rules that improves the perplexity, but it still doesn't use phrase boundary explicitly. PK-SCFG utilizes both restricted grammar and phrase boundary. For each model, initial values of $a(\beta|\alpha)$ were set uniformly and $b(w_c|\alpha), c(\beta w_f|\alpha)$ are set randomly.

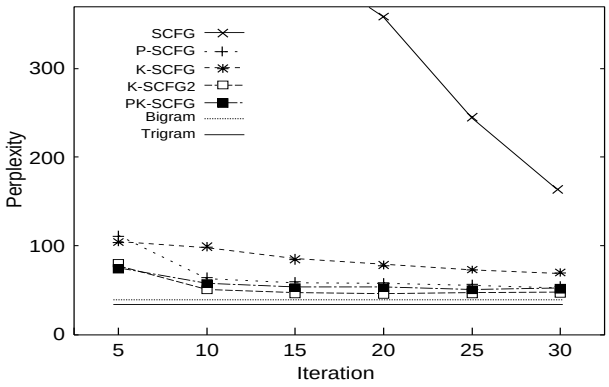

Figure 2: Comparison of various type of SCFGs

Figure 2 shows the experimental result of perplexity vs. iteration for each model. From this result, it was found that four enhanced models were much better than the original model. The perplexity of PK-SCFG was as well as that of K-SCFG2, that was the best model among five SCFGs. Figure 3 shows training times of each model. Enhanced models were much faster than the original SCFG. The use of restricted grammar made training process twenty times faster, and phrase boundary information made it eight times faster.

In the second experiment, PK-SCFG model was used as a language model of large vocabulary continuous speech recognition. The task domain was read speech of Japanese newspaper article from Mainichi Shimbun. The vocabulary size was 5000. The test set consisted of 100 sentences without any OOV words. The training set was 46301 sentences chosen from Mainichi Shimbun January to September of 1994, which contained no OOV words. Number of nonterminals was set to 100 and 120. The initial values of the models were set with two steps. In the first step, all words in the training sentences were replaced with its category name, and SCFGs were trained using that category name sequences. In the second step, output probability of

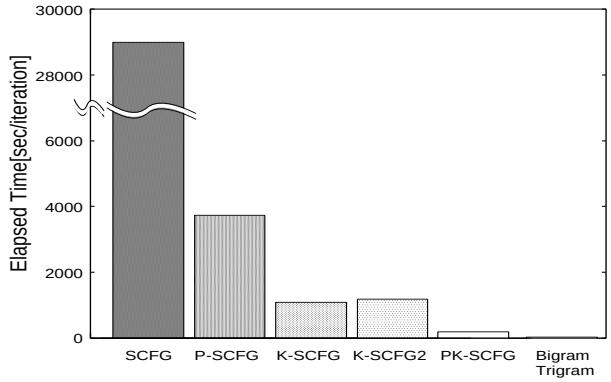

Figure 3: Elapsed time for estimating parameters of SCFGs

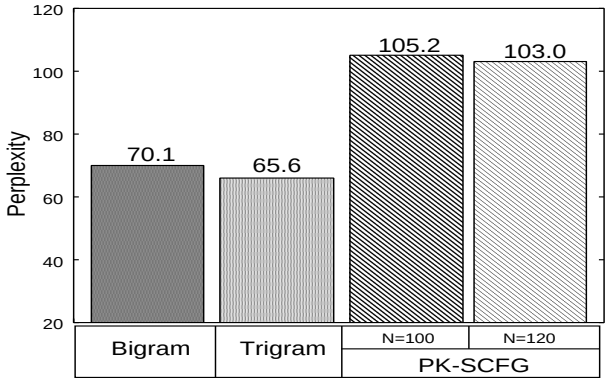

Figure 4: Perplexity of each model

Table 3: Optimum values of language model weights and insertion penalty

model	W_2	W_1	p
bigram	16	0	-20
trigram	17	0	-20
SCFG 100	0	19	-16
SCFG 120	0	18	-20
trigram+SCFG 100	14	10	0
trigram+SCFG 120	6	16	-12

a word was estimated as follows:

$$b'(\alpha \to w_c) = b(\alpha \to C(w_c))P(w_c|C(w_c)) \quad (15)$$

$$c'(\alpha \to \beta w_f) = c(\alpha \to \beta C(w_f))P(w_f|C(w_f)) \quad (16)$$

where $C(w)$ denotes the category name of w. Using these initial values, the models were trained again using the original training set.

Acoustic models in this experiments were HM-Nets with state clustering [3] which had 2000 states of 16 Gaussian mixture.

In this experiment, 100-best candidates were generated from input speech using bigram LM, then these candidates were rescored using trigram and PK-SCFG. The total score of a candidate W for input speech O was calculated as follows:

$$S(W|O) = W_1 \log P_1(W) + W_2 \log P_2(W) + S_a(O|W) + pn \quad (17)$$

where n was the length of the candidate, S_a was an acoustic score, P_1 and P_2 were probabilities from SCFG and n-gram respectively, W_1 and W_2 were language model weights of each model and p was an insertion penalty. Optimum values of those parameters are shown in Table 3.

Figure 4 shows perplexity of each model calculated upon test sentences. Perplexities of SCFGs are higher than that of bigram and trigram. Figure 5 shows word error rates obtained through rescoring. These result shows that SCFGs as good LM for LVCSR as bigram and trigram, and that the combination of trigram and SCFG achieves further improvement of WER.

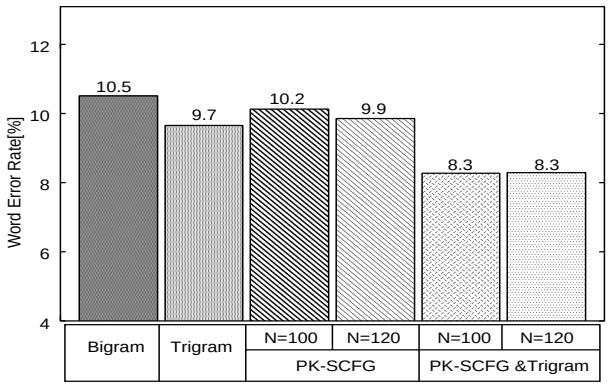

Figure 5: WER results

5. Summary

A new language model was proposed that utilizes dependency structure of Japanese language. This model is based on stochastic context free grammar and it reduces its computational complexity up to $O(N^2)$ by using restricted grammar. Word grouping using regular grammar is introduced for further reduction of training time. From the experimental result, the proposed model outperformed a traditional SCFG, and it gave lower WER on LVCSR task combined with trigram model.

6. REFERENCES

1. K. Lari and S. J. Young: "The estimation of stochastic context free grammars using the inside-outside algorithm", Computer Speech and Language, 4, pp. 35–56 (1990)
2. K. Lari and S. J. Young: "Application of stochastic context free grammars using the inside-outside algorithm", Computer Speech and Language, 5, pp. 237–257 (1991)
3. T. Hori, M. Katoh, A. Ito and M. Kohda: "A study on HM-Nets using decision tree-based successive state", Proc. ICSP97, pp. 383–387 (1997)