STUDIA PHONOLOGICA XXV (1991)

Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and *Quasi/Simplified*-Trigram

Seiichi NAKAGAWA, Isao MURASE, and Min ZHOU

ABSTRACT

In this paper, we investigate the language models by stochasic context-free grammar (SCFG), bigram and *quasi*-trigram. For calculating of statistics of bigram and *quasi*-trigram, we used the set of sentences generated randomly from CFG that are legal in terms of semantics. We compared them on the perplexities for their models and the sentence recognition accuracies. The sentence recognition was experimented in the "UNIX-QA" task with the vocabulary size of 521 words. From these results, the perplexities of bigram and *quasi*-trigram were about 1.6 times and 1.3 times larger than the perplexity of CFG that corresponds to the most restricted grammar (perplexity=10.0), and the perplexity of SCFG is only about 1/2 of CFG. We realized that *quasi*-trigram had the almost same ability of modeling as the most restricted CFG when the set of plausible sentences in the task was given.

1. INTRODUCTION

In continouus speech recognition, the language processing can improve the recognition accuracy by correcting the recognition errors occurred by using only acoustic feature, because it can use high-order knowledges such as syntax, semantics, pragramtics and so on. As the typical languaeg models, the models such as trigram, regular grammar, context-·ree grammar (CFG), augmented transition network (ATNG), dependence grammar (kakari-uke) have been well-known.

The CFGs have been used for the part of language processing in the sentence rocognition system from the facts that are suitable for the natural language models and have well-known efficient parsing methods. However, it is difficult to construct the CFG that accepts only the sentences allowed for the given tasks, in particular, on the case of conversational sentences.

The bigram and trigram are stochastic grammars that approximate the sentence (word sequence) occurrence probabilities using the probabilities of the word-

Seiichi NAKAGAWA: (中川聖一): Professor, Department of Information and Computer Sciences, Toyohashi University of Technology, Tempaku-cho, Toyohashi, 441 Japan

Isao MURASE (村瀬 功): Graduate Student, Department of Information and Computer Sciences, Toyohashi University of Technology, Tempaku-cho, Toyohashi, 441 Japan

Min ZHOU (周 旻): Graduate Student, Department of Information and Computer Sciences, Toyohashi University of Technology, Tempaku-cho, Toyohashi, 441 Japan

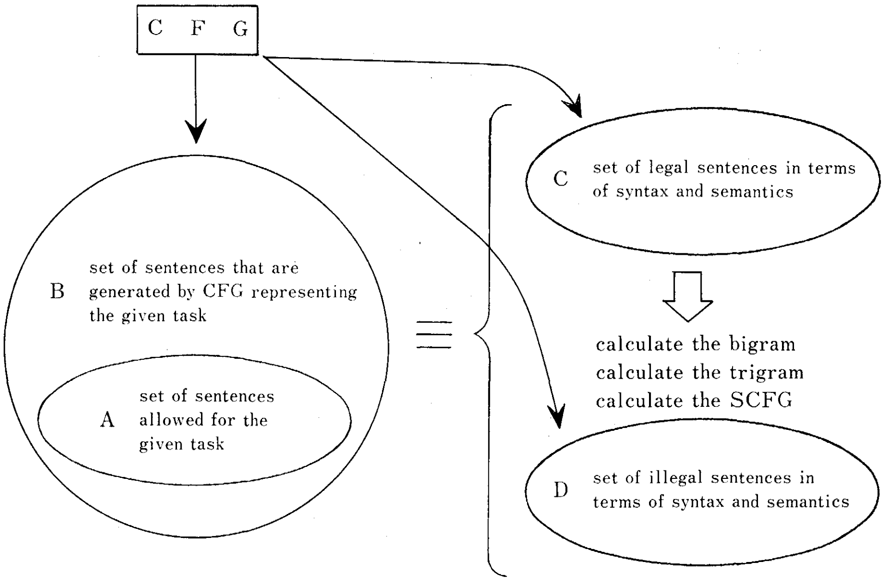
The authors have done the research under the direction of Dr. Shuji Dohshita, Professor of Information Science, Kyoto University

pair or word-trio. Usually, these probabilities of the word-pair or word-trio are calculated statistically from the training sentence set. In short, even if the given task is changed, it is easy to construct the bigram or trigram model. On the other hand, it is difficult to construct the CFG automatically from the given sentence set.

We have experimented on the sentence ıecognition using bigram and *quasi*-trigram[1],[8]. However, the statistics of bigram and *quasi*-trigram were calculated from all possible sentences generated by CFGs, so these could not inevitably approximate the given CFGs. Since the perplexities of bigram and *quasi*-trigram were about 2.6 times and 2 times larger than that of given CFG's, however, we have expected that the stochastic grammars have the descriptive ability as same as the most restricted CFG, if the stochastic grammars were estimated from only the set of sentences that were correct in terms of syntax and semantics. This is caused by the fact that the CFGs generate many illegal sentences in terms of semantics.

In this paper, we compare the bigram and *quasi*/*simplified*-trigram that are constructed by the set of legal sentences with the CFG constructed for the given task. For this purpose, we must prepare the set of thousands of sentences. Since it is difficult to make the set by handiwork, however, we have made the set by picking out only the legal sentences in terms of semantics from the sentences that are generated randomly by CFG constructed in advance. We compared their models in terms of perplexities and sentence recognition rates. We also constructed a SCFG from the legal sentences.

We illustrate the outline of the experiment way in Fig. 1. The sentence set A allowed for the given task is almost equal to the sentence set C which sentences are generated from CFG and are acceptable in terms of syntax and semantics. In fact, though we use a part of its set, the grammar estimated using this partial set could


C F G
B
set of sentences that are generated by CFG representing the given task
A
set of sentences allowed for the given task
≡
C
set of legal sentences in terms of syntax and semantics
calculate the bigram
calculate the trigram
calculate the SCFG
D
set of illegal sentences in terms of syntax and semantics

$A \subset B,\ B = C \cup D,\ A \fallingdotseq C$

Fig. 1. The relationship among sentence sets

generate almost all sentences in the set A. This paper focuses on comparing the stochastic grammar estimated from the set C with the CFG. We approximate the grammar of the set C with bigram and trigram.

2. The MEASUREMENT OF TASK COMPLEXITY

2.1 *Perplexity*[2]

The entropy of CFG is calculated by the following equation by calculating the distribution of the sentence length from training sentences and assuming that sentences with the same length occur with a uniform probability distribution (The length denotes the number of words or phoneme in a sentence):

P_k: occurrence probability of the sentence with length k.

N_k: the number of all sentences with length k in language L.

Therefore, the entropy of language L is defined as follows:

$$H_0(L) = -\sum_{w_1^k} P(w_1^k) \log_2 P(w_1^k) = -\sum_{w_1^k} \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k}$$

$$= -\sum_k N_k \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} = -\sum_k P_k \log_2 \frac{P_k}{N_k}.$$

The entropy per unit is

$$H(L) = -\sum_{w_1^k} \frac{1}{k} \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} = -\sum_k \frac{P_k}{k} \log_2 \frac{P_k}{N_k}.$$

The perplexity is defined as the following equation:

$$F_p(L) = 2^{H(L)}.$$

In the case of bigram and trigram the entropies are described by the following equations:

$$H(L) = -\sum P(w_i w_j) \log_2 P(w_j \mid w_i): \qquad \text{bigram,}$$

$$H(L) = -\sum P(w_i w_j w_k) \log_2 P(w_k \mid w_i w_j): \qquad \text{trigram.}$$

The trigram that we used in practice is the *quasi*-trigram that the head word of the word-trio is transformed to one of sub-classes because of enormous combinations of word-word-word. In other words, the trio is subclass-word-word or subclass-subclass-subclass. Here, we classified the words into sixteen categories by Japanese part of speech. The categories of parts of speech are the followings:

1. normal noun
2. pronoun
3. proper noun 1
4. proper noun 2
5. sa-hen noun
6. numeral
7. verb
8. auxiliary verb
(connect with sa-hen noun)
9. adjective relative
10. conjunction
11. postfix
12. adjective
13. adverb
14. prefix
15. auxiliary verb
16. particle (postposition)

Since there exist some words belonging to several categories, however, we can not judge which categories each word belongs to. Therefore we classify the word class to subclass. Here, the word class is the class that consists of some resemble words in terms of semantics. We used 268 classes. The CFG used here treats this word class as a terminal symbol, not used a word. Therefore we assume that the words in a word calss occur with a uniform probability distribution.

In the case of *quasi*-trigram, the probability $\mathrm{P}(\mathrm{w}_k|\mathrm{w}_i\mathrm{w}_j)$ in the above equation of the entropy was calculated by the following equation, assuming that the words in each subclass occur with a uniform probability distribution. Here, $|\mathrm{s}_i|$ denotes the number of words in subclass s_i and $\mathrm{H}(\mathrm{w}_i\mathrm{w}_j\mathrm{w}_k)$ denotes the cooccurrence frequency of the trio.

$$P(w_k|w_iw_j)=\frac{H(w_iw_jw_k)}{H(w_iw_j)}$$
$$=\frac{\sum_{i:w_i\in s_i} H(s_iw_jw_k)/|s_i|}{\sum_{i:w_i\in s_i} H(s_iw_j)/|s_i|}.$$

And, the probability $\mathrm{P}(\mathrm{w}_j|\mathrm{w}_i)$ is calculated by the following equation to obtain the word unit entropy from word class sequences. Here, $|\mathrm{c}_i|$ is the number of words in word class c_i, $\mathrm{H}(\mathrm{c}_i)$ and $\mathrm{H}(\mathrm{c}_i\mathrm{c}_j)$ means the occurence frequence of c_i and $\mathrm{c}_i\mathrm{c}_j$, respectively.

$$P(w_j|w_i)=\frac{H(w_iw_j)}{H(w_i)}$$
$$=\frac{\sum_{i:w_i\in c_i,j:{}_jw\in c_j} H(c_ic_j)/(|c_i|\times|c_j|)}{\sum_{i:w_i\in ci} H(c_i)/|c_i|}.$$

In the case of *quasi*-trigram, the probability $\mathrm{P}(\mathrm{w}_k|\mathrm{w}_i\mathrm{w}_j)$ is calculated from the occurence frequences of $\mathrm{H}(\mathrm{s}_i\mathrm{c}_j\mathrm{c}_k)$ and $\mathrm{H}(\mathrm{s}_i\mathrm{c}_j)$ as following:

$$P(w_k|w_iw_j)=\frac{H(w_iw_jw_k)}{H(w_iw_j)}$$
$$=\frac{\sum_{i:w_i\in s_i,j:w_j\in c_j,k:w_k\in c_k} H(s_ic_jc_k)/(|s_i|\times|c_j|\times|c_k|)}{\sum_{i:w_i\in s_i,j:w_j\in c_j} H(s_ic_j)/(|s_i|\times|c_j|)}.$$

2.2 *Test set perplexity*[3]

Generally, the perplexity must be calculated for the set of test sentences, because the relative difficulty of sentence recognition depends on test sentences when the number of test sentences is small. We call it "test set perplexity". For removing the necessity of both probabilities of the start and end of sentence for sentence recognition process, we defined that the sentence begins or ends with a mark of a sentence boundary. If the word sequence for a given sentence is represented by "$\sqcup \mathrm{w}_1\mathrm{w}_2\cdots \mathrm{w}_n \sqcup$", the test set perplexity is defined as follows:

- General formula

$$F_T(p)=\left(\frac{1}{P(w_1\,|\,\sqcup)}\times\frac{1}{P(w_2\,|\,\sqcup w_1)}\times\frac{1}{P(w_3\,|\,\sqcup w_1 w_2)}\right.$$
$$\left.\times\cdots\times\frac{1}{P(w_n\,|\,\sqcup w_1\,w_2\cdots w_{n-1})}\times\frac{1}{P(\sqcup\,|\,\sqcup w_1\,w_2\cdots w_n)}\right)^{1/(n+1)}$$

- Bigram

$$F_T(p)=\left(\frac{1}{P(w_1\,|\,\sqcup)}\times\frac{1}{P(w_2\,|\,w_1)}\right.$$
$$\left.\times\cdots\times\frac{1}{P(w_n\,|\,w_{n-1})}\times\frac{1}{P(\sqcup\,|\,w_n)}\right)^{1/(n+1)}$$

- Trigram

$$F_T(p)=\left(\frac{1}{P(w_1\,|\,\sqcup\sqcup)}\times\frac{1}{P(w_2\,|\,\sqcup w_1)}\right.$$
$$\left.\times\cdots\times\frac{1}{P(\sqcup\,|\,w_{n-1}w_n)}\times\frac{1}{P(\sqcup\,|\,w_n\sqcup)}\right)^{1/(n+1)}$$

In the case of *quasi*-trigram or wordclass-based bigram, we calculate the test set perplexity using probabilities $\mathrm{P}(\mathrm{w}_k|\mathrm{w}_i\mathrm{w}_j)$ and $\mathrm{P}(\mathrm{w}_j|\mathrm{w}_i)$ described in section 2.1. When there are several test sentences, we calculate the geometrical mean of the test set perplexity for each test sentence.

In the speical case that not consider the occurrence probability but consider only the number of words predicted (branching number), if the number of the predicted words in the right side of $\mathrm{w}_1\,\mathrm{w}_2\cdots\mathrm{w}_{i-1}$ is c_i, then the test set perplexity is the following:

$$F(p)=(c_1c_2\cdots c_n)^{1/n}.$$

In short, it is a geometrical mean of branching numbers.

3. Language Models

3.1 *Task*[4],[9]

The experiments were implemented in the "UNIX-QA" task with the vocabulary size of 521 words and perofrmed with fifty Japanese test sentences (the average length is 7.76 words/sentence) for every six male speakers.

3.2 *Context-free grammars*

The grammars used here are the perplexities of 10.0, 15.9, 19.3, 25.4 and 50.7 and all of them are represented by CFG[4]. For the convenience sake, their grammars are denoted by A, B, C, D and E, respectively. The differences are the following:

A: semantic grammar depending on the task
B: case grammar restricting the noun phrase at the front of the postpoisitions
C: case grammar loosing the restriction in B

D: case grammar removing the restriction in B

E: grammar without semantic information

Because the most restricted grammar A generates more than 10^{37} sentences, the majority of these sentences are illegal in terms of syntax and semantics. Each grammar can generate all sentences that are allowed by the given task. The grammar A includes a set of 535 rewriting rules (the number of non-terminal symbols of 259) and 600 rewriting rules that produce words from word classes.

3.3 *Stochastic Grammars*

3.3.1 Sentence sets

To construct the bigram or *quasi/simplified*-trigram model, the set of sentences concerning with the given task is necessary. The sentence set used here is a set of legal sentences that are selected out of sentences generated randomly by the grammar A of CFG in terms of semantics. The way of sentence generation is performed by outputting a word or a nonterminal symbol using random numbers at each branching point when the CFG is expanded from top to down. Therefore there is the inclination of the branching number followed by branching at each branching point. So we could not generate all sentences with a uniform probability distribution and some sentences were frequently generated several times. In our example, the most generated sentence has the occurrence probability of 0.058. This means the sentence appears 58 times when 1000 sentences are generated. However it is difficult to generate all sentences with a uniform probability distribution from the grammar.

We made 3 types of sentence sets. One of these has 2048 sentences that are legal in terms of syntax and consists of about 50% of sentences generated randomly. The second one has 3000 sentences that are legal in terms of semantics and consists of about 25% of sentences generated randomly. The last one has 5048 sentences that disjunct the first two sets. They are denoted by "sent1", "sent2" and "sent_mix", respectively.

Since each set contains duplicated sentences, the number of different sentences is 1678, 2193 and 3624, respectively, when the duplicated sentences are removed. In either case, we can not construct the bigram or *quasi*-trigram model by the reason of insufficient samples.

Accordingly we transformed the sentences (word sequences) of each set to the word class sequences. In this way, many sentences are re-generated from a word class sequence, therefore we can enlarge the small set to the large set in appearance. For example, if the sentence "Mr. A will receive the mail" is generated, the sentence "Miss B receives the mail" may also be generated. The number of sentences generated from each set of word class sequences is shown in Table 1 (there are no duplicated sentences).

3.3.2 Bigram

The perplexities for 6 kinds of sentence sets mentioned in section 3.3.1 are shown in Table 2. Table 2 (a) is on the case using no flooring method for an extremely small probability, and Table 2 (b) is on the case using a flooring method. This

Table 1. The number of training sequences

	sent1	sent2	sent_mix
with duplicated sentences	2048	3000	5048
without duplicated sentence	1678	2193	3624
word sequences generated from wordclass sequences	6.5×10^9	2.4×10^8	6.7×10^9

Table 2 The word unit perplexities for bigram

(a) using no flooring method

	perplexity			test set perplexity			
				perplexity1		perplexity2	
	word	0/1	class	word	0/1	word	0/1
sent1	7.8	19.4	4.7	11.1	18.3	14.8	21.5
sent2	7.1	17.4	4.5	12.0	17.7	16.0	20.8
sent_mix	7.6	20.5	4.8	11.7	19.5	15.6	22.9

(b) using flooring method (flooring value = 0.05)

	perplexity			test set perplexity			
				perplexity1		perplexity2	
	word	0/1	class	word	0/1	word	0/1
sent1	18.3	–	12.3	14.7	–	19.8	–
sent2	14.2	–	9.7	13.8	–	18.4	–
sent_mix	11.9	–	7.8	12.7	–	16.9	–

flooring method is that replaces the frequency of 0 with 0.05. This value was decided by the view point that the test set perplexity and the perplecity of training data set should become equivalent. We use no flooring method in the experiments, because all of 50 test sentences (representative sentences concerning with the given task) are generated and the recognition accuracies become to be poor by considering the word-pairs that generally not occur if the flooring method was used.

The column of "word" in each table denotes the perplexities when the word class sequences were transformed to the word sequences, the "0/1" is the case of word-pair without probabilities, and the "class " is the perplexities of word class unit. The "perplexity1" and "perplexity2" correspond whether the probabilities of end of sentences (or word boundary mark) are included or not. We have not used the probabilities of end of sentences in the sentence recognition process and used only the information whether the word is the end of sentence or not, therefore we use "perplexity2" as the test set perplexity hereafter.

From these tables, we understand that the test set perplexities are about 16. For these things, these perplexities are about 1.6 times larger than the perplexity of the most restircted CFG (perplexity=10.0).

The reason, that the word class unit perplexities in the "class" columns are smalll about 5, is that the number of word classes is only 268, just only about half of 521 words.

3.3.3 *Quasi-trigram*

In the same way of bigram, we constructed the *quasi*-trigram model and cal-

Table 3 The word unit perplexities for *quasi*-trigam

(a) using no flooring method

	perplexity			test set perplexity			
				perplexity1		perplexity2	
	word	0/1	class	word	0/1	word	0/1
sent1	4.1	10.0	3.3	9.1	13.2	12.1	15.5
sent2	3.9	9.7	3.1	9.8	12.5	12.8	14.6
sent_mix	4.1	10.9	3.3	9.5	14.0	12.5	16.2

(b) using flooring method (flooring value = 0.01)

	perplexity			test set perplexity			
				perplexity1		perplexity2	
	word	0/1	class	word	0/1	word	0/1
sent1	17.3	–	19.7	12.6	–	17.1	–
sent2	12.7	–	13.7	11.2	–	14.7	–
sent_mix	8.9	–	8.9	10.1	–	13.3	–

Table 4 The word unit perplexities for *simplified*-trigram

	test set plerplexity			
	perplexity1		perplexity2	
	word	0/1	word	0/1
sent1	10.2	–	13.7	–
sent2	11.7	–	15.9	–
sent_mix	11.0	–	14.9	–

culated the perplexities fcr each sentence sets. Though the number of word calss-trio is 16×268×268=1149184, every estimated *quasi*-trigram models could generate all test sentences even if the flooring method was not used. The perplexities are shown in Table 3. The flooring value at this time is 0.01.

From these tables we could know that the test set perplexities are about 13. Therefore these perplexities are about 1.3 times larger than that of the most restricted CFG.

Also, the perplexities on the case using occurrence probabilities are 20~30% smaller than the case using only the binary information whether the pair or trio exists or not.

3.3.4 *Simplified-trigram* (*Extended*-bigram)

Also, we make the bigram using the probabilities whether the "$w_i \square w_j$" occurs or not, where "□" denotes an arbitrary word. And, using this model with the normal bigram together, called *simplified*-trigram or *extended*-bigram, we experimented the sentence recognition.

The probability $P(w_k|w_i w_j)$ for the bigram is the product of the probabilities of normal bigram $P(w_k|w_j)$ and bigram $P(w_k|w_i)$. Of course, these probability should be normalized, that is,

$$P(w_k|w_i w_j)=\alpha \cdot P(w_k|w_j) \cdot P(w_k|w_i),$$

$$\alpha=\frac{1}{\sum_k P(w_k|w_j) \cdot P(w_k|w_i)}.$$

We calculated the perplexities using both probabilities of normal bigram and *jumped*-bigram for each set. The test set perplexities are shown in Table 4.

From this table we know that the test set perplexities are about 14. Therefore these perplexities are about 1.4 times larger than that of the most restricted CFG.

3.3.5 Stochastic Contest-Free Grammar[7]

The stochastic context-free grammar model was estimated by a method of using CKY parsing. The initial parameters for the SCFG was taken from the CFG with a random piobabilities. For the use of the CKY parsing algorithm, the original CFG was transformed into the Chomsky normal form.

Based on CKY parsing, every sentences in training set were analysized and each rule used to generate the sentences also be countted. The larning of the probabilities attached to rules is based on counting the times of each rule used in parsing the sentence set. It will enable us to estimate the rule probabilities

$$\{\mathrm{Prob}(\alpha\rightarrow\beta)\},$$

which means the probabilities that the non-terminal symbol α at the left side will be replaced with the right side patterns β. When the SCFG is in the Chomsky normal form, α is a non-terminal and β is two successive symbols of non-terminal or one terminal.

For each sentence B(i) in a training set, based on CKY parsing it could be analisized into one or more parsing trees if it is ambiguous. Denoting D(i,j) as the j-th parsing tree of the sentence B(i), and num$(\alpha\rightarrow\beta)$ as the number of times that the rule $(\alpha\rightarrow\beta)$ is appeared in D(i,j), then the probability of parsing tree D(i,j) could be calculated as

$$\mathrm{Prob}\ (\mathrm{D(i,j)})=\prod \mathrm{Prob}\ (\alpha\rightarrow\beta).$$

and the times of rule $(\alpha\rightarrow\beta)$ used to parse the sentence B(i) is given by

$$\mathrm{C}(\mathrm{i},\alpha\rightarrow\beta)=\frac{\sum_j \mathrm{Prob}\ (\mathrm{D(i,j)})\times \mathrm{num}\ (\alpha\rightarrow\beta)}{\sum_k \mathrm{Prob}\ (\mathrm{D(i,k)})},$$

After normalization of $\mathrm{C}(\mathrm{i},\alpha\rightarrow\beta)$

$$\mathrm{f}(\alpha\rightarrow\beta)=\frac{\sum_i \mathrm{C}(\mathrm{i},\alpha\rightarrow\beta)}{\sum_i\sum_\gamma \mathrm{C}(\mathrm{i},\alpha\rightarrow\gamma)},$$

it is replaced with the Prob$(\alpha\rightarrow\beta)$. Repeating the iteration until parameters be converged will bring us an estimated SCFG. From the estimating process, obviously the rule frequently used has a high probability and the rarely used one has a lower probability.

By using of SCFG model, the perplexities of training sentence and test sentence sets are listed in Table 5. Both are about 1/2 of the perplexity of the original CFG.

Here we should notice that we used only legal sentences which were generated from the original CFG.

Table 5 The word unit perplexities of SCFG (beam width=45)

	sample data		test data	
	word	class	word	class
sent_mix	4.53	2.95	5.71	3.43

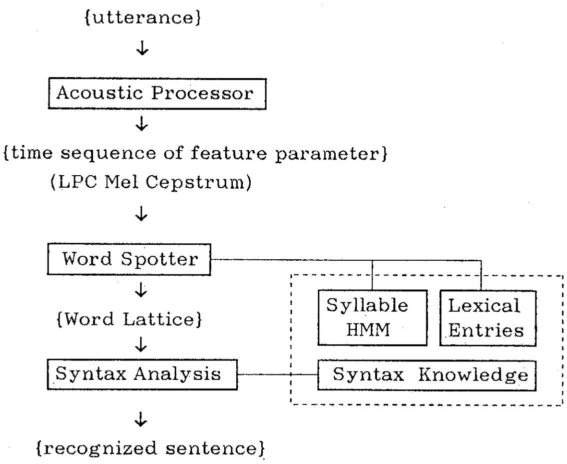


Fig. 2. System organization of SPOJUS-SYNO

4. Recognition Experiments

4.1 *Recognition System and Speech Materials*

Figure 2 illustrates the recognition system (SPOJUS-SYNO). At firstl this system makes word-based HMMs automatically by concatenating syllable-based (trained) HMMs. Japanese syllables consist about 110 syllables each of which is composed of a consonant and a vowol(CV), a syllabic nasal(N), a vowol(V), or a consonant, a semivowol and a vowel(CYV). We adopted a continuous output probability HMM with a discrete duration probability[4],[10]. This model consists of five states and four transitions. The four parameters set of duration (transition) and output probabilities (the mean vector and covariance matrix of feature vectors) were calcuated by suing the Baum-Welch estimation algorithm. Then a word lattice is hypothesized by a word spotting algorithm and word-based HMMs. A hypothesized word consists of the beginning frame, the ending frame, the matching score (likelihood) and the word name. Finally, the time-synchronous left-to-righ parsing algorithm is executed to find the best word sequence from the word lattice according to syntactic and semantic knowledge represented by a context-free semantic grammar, bigram, trigram and so on.

The recognizer makes the partial sentence hypotheses frame-synchronously. The parser predicts possible words at the righ-hand side for each given partial sentence hypothesis at the current processing frame. The partial sentence hypotheses are updated by connecting predicted candidate word in the word lattice. This process is continued until the ending frame of the input sentence. All experiments were performed in a multi-speaker mode[4].

Each of six male speakers uttered 216 words, 80 loan (foreign) words and 50 sentances in a soundproof room, respectively. These words were segmented into syllable units by the inspection and used for training syllable-based HMMs. The other fifty sentences for test data were related to the content of "Question or Demand for Electric Mail", which was a part of the task of UNIX-QA. The speed of utterances ranged from 8 to 9 morae per second (about 16 to 18 phonemes per second).

It was moderately fast. One sentence consists of 7.4 words on the average. These utterances were sampled/digitized with the accuracy of 12 bits/sampling by 12 kHz and analyzed by the 14 order LPC. We obtained 14 LPC cepstrum coefficients and signal power for every 5 ms. These coefficients were transformed to 10 LPC mel-cepstrum coefficients. The vocabulary size of a part of the task is 521 words.

4.2. *Sentence recognition results*[8]

The recognition results using 5 kinds of CFGs are shown in Table 6. The estimation values in this table are the results estimated by using the evaluation method we proposed[5]. The results for the bigram constructed from sent2 (not use the flooring method) are shown in Table 7, the case of *quasi*-trigram constructed from sent_mix (not use flooring method) is shown in Table 8 and the case of *simplified*-trigram constructed from sent_mix (not use flooring method) is shown in Table 9. Finally the case of the stochastic context-free grammar is summarized in Table 10.

Table 6 Sentence recognition results using CFG (beam width=45)

perplexity	10.0		15.9		19.3		25.4		50.7	
speaker	sent.rec.%		sent.rec.%		sent.rec.%		sent.rec.%		sent.rec.%	
	est.	exp.	est.	exp.	est.	exp.	est.	exp.	est.	exp.
TI	76.5	78.7	66.1	66.0	61.4	63.8	54.6	63.8	37.7	51.1
MA	54.6	79.6	40.8	46.9	35.2	44.9	27.8	34.7	14.1	26.5
HU	68.3	71.7	56.5	58.7	51.9	54.3	44.1	45.7	27.8	32.6
KO	67.2	71.7	55.1	58.7	49.8	56.5	43.4	47.8	26.3	26.1
SE	77.7	60.0	70.5	51.1	66.1	51.1	59.9	40.0	42.6	20.0
SN	60.9	50.0	46.1	25.0	40.8	25.0	31.6	18.8	14.5	8.3
average	67.5	68.7	55.9	51.1	50.9	49.3	43.6	41.8	27.2	27.4

Table 7 Tecognition results using bigram for sent 2 (beam width=100)
(test set perplexity2 = 16.0)

speaker	input	word acc.	% correct	subst.	ins.	del.	sent. rec.
TI	341	91.2%	90.3%	6.7%	0.9%	2.1%	61.7%
MA	372	84.4	82.8	12.9	1.6	2.7	51.0
HU	328	87.5	85.1	10.7	2.4	1.8	56.5
KO	334	88.3	86.5	9.6	1.8	2.1	45.7
SE	319	86.2	83.4	12.5	2.8	1.3	44.4
SN	357	73.4	64.4	22.7	9.0	3.9	51.0
average	342	85.2	82.1	12.5	3.1	2.3	47.3

Table 8 Recognition results using *quasi*-trigram for sent_mix (beam width=100)
(test set perplexity2 = 12.5)

speaker	input	word acc.	% correct	subst.	ins.	del.	sent. rec.
TI	341	95.0%	94.7%	3.8%	0.3%	1.2%	76.6%
MA	372	87.1	85.5	10.8	1.6	2.2	53.1
HU	328	87.5	86.3	10.4	1.2	2.1	56.5
KO	334	87.7	85.6	9.9	2.1	2.4	52.2
SE	319	88.1	87.2	10.7	0.9	1.3	46.7
SN	357	73.7	68.1	23.8	5.6	2.5	29.2
average	342	86.5	84.6	11.7	2.0	2.0	52.4

Table 9 Recognition results using *simplified*-trigram for sent_mix

(beam width=100) (test set perplexity2 = 14.9)

speake	input	word acc.	% correct	subst.	ins.	del.	sent. rec.
TI	341	94.7%	93.6%	4.4%	1.2%	0.9%	70.2%
MA	372	88.4	87.6	9.7	0.8	1.9	61.2
HU	328	92.1	90.6	6.1	1.5	1.8	65.2
KO	334	91.0	90.1	6.9	0.9	2.1	63.0
SE	319	91.5	90.6	7.8	0.9	0.6	57.8
SN	357	72.0	59.1	25.2	12.9	2.8	27.1
average	342	88.3	85.3	10.0	3.0	1.7	57.3

Table 10 Recognition result using SCFG (beam width=45)

speaker	TI	MA	HU	KO	SE	SN	average
sentence recognition rate (%)	78.7	78.6	73.3	69.6	62.2	54.2	69.5

Since the perplexity of the bigram in Table 7 is 16.0, this is compared with the results for the CFG (grammar B) with the perplexity of 15.9. Also the perplexity of the *quasi*-trigram in Table 8 is 12.5, so this is compared with the results for the CFG with the perplexity of 10.0 or 15.9 (grammar A or B). And the perplexity of the *simplified*-trigram in Table 9 is 14.9, so this is compared with the results for the CFG with the perplexity of 15.9 (grammar B). In either case the corrdspoding sentence recognition rates are nearly equal. In short, if the perplexities are equal each other, the recognition rates are equal even if any language models are used[1]. On the other hand, we expected that the using case of SCFG would remarkable improve the recognition rate, however, the rate was slightly improved. We guess one of the reasons that the hypothesized sentences (recogniton results) using CFC were almost all legal sentences on the syntax and semantics. We have an open problem for combining the time synchronous recognition algorithm with SCFG.

Since the perplexity is 12.5 on the case of the *quasi*-trigram, the model is roughly equivalent to the CFG with the best accuracy. And we understand that the *quasi*-trigram model has the descriptive ability as same as the most restricted CFG. Therefore the *quasi*-trigram model has higher ability for the language modeling. On the other hand, the *simplified*- trigram has the descriptive ability as almost same as the *quasi*-trigram because the perplexities is 14.9, and the estimation of *simplified*-trigram is easier than that of *quasi*-trigram.

When the task is changed, it is difficult to construct the CFG with the good accuracy that generate or accept only the legal sentences allowed by the given task and not generates or accepts the other illegal sentences. But the trigram model has the advantage that the models with the good accuracy are constructed briefly if the large number of training sentences is given.

4.3. *Consideration on processing time*

For one sentence, the number of spotted words is abot 5000, and the processing times are 27 minutes for bigram, 28 minutes for *quasi*-trigram and 13 minutes for C-FG (in the case of the grammar A: perplexity=10.0) using SUN-SPARK station-1

(12 MIPS). The processing time is usually expected that the fastest one is trigram and next one is bigram and the latest is CFG, but the order became to opposite. The reason is as follows:

First, although the *quasi*-trigram model calculates the occurrence frequencies of subclass-wordclass-wordclass before the parsing process begins, the probabilities of subclass-word-word are calculated at every time that the sentence hypotheses are updated (because it can not take the array size of 16×521×521). For these things, the processing time of the *quasi*-trigram model becomes to be later.

Second, the perplecity of *quasi*-trigram is slightly larger than that of CFG. Third, the sentence hypotheses are expanded and pruned by the beam search strategy for keeping off the explosively increase of the number of sentence hypotheses. Since the beam search width is 45 for CFG, 100 for bigram and 100 for *quasi*-trigram, the CFG becomes to be the fastest one.

In short, for the bigram the beam width is about 2 times and the perplexity is about 1.5 times larger than the case of CFG, however, the processing time is about 2 times. Therefore we can guess that the processing time for the language model by the bigram is slightly faster than that by CFG for one partial sentence hypothesis. Since the *quasi*-trigram model has smaller perplexity than that of the bigram, the processing time will be faster than the bigram. However, in practice, the beam search width depends on the perplexity.

5. CONCLUSION

In this paper, we investigated the language models for the bigram and *quasi*-trigram and SCFG. Constructing each model from sentence set, the perplexities of bigram and *quasi*-trigram were about 1.6 times and 1.3 times larger than that of the most restricted CFG's, respectively. Though we uses the *quasi*-trigram that the head word of the word-trio is transformed to one of sub-classes, the descriptive ability will become to be higher if we use a normal trigram. The *simplified*-trigram has the perplexities as almost same as the *quasi*-trigram. From these results we can conclude the trigram model has the high ability for language model because the descriptive ability of trigram model is equivalent to the most restricted CFG.

When the task applied to the recognition system is changed, it is difficult to construct the restricted CFG that generates only the legal sentences allowed for the given task and not the others. However, the bigram and trigram models are constructed easily when the set of sentences concerning to the given task is given, because these use only the statistics. But we should notice that it is also difficult to prepare the set of sentences that are more than thousands. The superiority or inferiority strongly depends on the amount of available data base and the complexity of a given task (vocabulary size, perplexity and so on).

REFERENCES

[1] I. Murase and S. Nakagawa.: "Sentence Recognition Method Using Word Cooccurrence Probability and Its Evaluation", Proc. ICSLP90, Vol. 2, pp. 1217–1220 (1990)

[2] Y. Niimi et al.: "Indices to Measure the Complexity of Speech Recognition Tasks", 2nd Symp.

Advanced Man-Machine Interface Through Spoken Language (1988)

[3] K-F. Lee: "Large-Vocabulary Speech Independent Continuous Speech Recognition: The SPHINX System", Ph. D. thesis, Carnegie-Mellon University (1988)

[4] S. Nakagawa, et al.: "Syntax Oriented Spoken Japanese Recognition/Understanding System-SPOJUS-SYNO—", IEICE Trans. Vol. J72-D-II, p. 1276–1283 (1989, in Japanese)

[5] S. Nakagawa, et al.: "An Evaluation Method for Continuous Speech Recognition Systems-Relationship between Task Complexity and Sentence Recognition Accuracy", IEICE Trans. Vol. J73-D-II, pp. 683–693 (1990), (in Japanese)

[6] S. Nakagawa.: "Speaker-Independent Continuous-Speech Recognition by Phoneme-Based Word Spotting and Time-Synchronous Context-Free Parsing", Computer Speech & Language, Vol. 3, No. 3, pp. 277–299 (1989)

[7] T. Fujisaki, F. Jelinek, J. Cocke, E. Black, T. Nishino. A Probabilistic Parsing Methode for Sentence Disambiguation, Proc International Parsing Workshop'89, pp. 85–94 (1989)

[8] S. Nakagawa and I. Murase: "Comparison of Language Models by Context-Free Grammar, Bigram and Quasi/Simplified-Trigram", IEICE Trans, Vol. E74, No. 7, pp. 1897–1905 (1991)

[9] S. Nakagawa et al.: "Comparison of Syntax-Oriented Spoken Japanese Understanding System with Semantic-Oriented System," IEICE Trans, Vol. E74, No. 7, pp. 1854–1862 (1991)

[10] S. Nakagawa and Y. Hirata.: "Comparison Among Time-delay Neural Networks, LVQ2, Discrete Parameter HMM and Continuous Parameter HMM", Proc. ICASSP, pp. 509–512 (1990)