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 stochastic 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 continuous 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, pragmatics and so on. As the typical language models, the models such as trigram, regular grammar, context-free 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 recognition 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-
Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and Quasi-Simplified-Trigram

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 recognition using bigram and quasi-trigram\cite{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

\[ A \subset B, \quad B = C \cup D, \quad A \not\subset C \]

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\textsuperscript{[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:

\[
H_0(L) = -\sum \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} = -\sum \frac{P_k\log_2 P_k}{N_k}.
\]

The entropy per unit is

\[
H(L) = -\sum \frac{1}{N_k} \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} = -\sum \frac{P_k\log_2 P_k}{N_k}.
\]

The perplexity is defined as the following equation:

\( P_p(L) = 2^H(L). \)

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

\[
H(L) = -\sum P(w_i|w_j) \log_2 P(w_i|w_j): \text{ bigram,}
\]

\[
H(L) = -\sum P(w_i|w_j|w_k) \log_2 P(w_i|w_j|w_k): \text{ trigram.}
\]

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
9. adjective relative
10. conjunction
11. postfix
12. adjective
13. adverb
14. prefix
15. auxiliary verb
16. particle (postposition)

(connect with sa-hen noun)
Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and Quasi/Simplified-Trigram

Since there exist some words belonging to several categories, however, we cannot 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 class occur with a uniform probability distribution.

In the case of quasi-trigram, the probability $P(w_k|w_i,w_j)$ 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, $|s_i|$ denotes the number of words in subclass $s_i$ and $H(w_i,w_j,w_k)$ denotes the cooccurrence frequency of the trio.

$$P(w_k|w_i,w_j) = \frac{H(w_i,w_j,w_k)}{H(w_i,w_j)}$$

$$= \frac{\sum_{i : w \in s_i} H(s_i,w_j,w_k)/|s_i|}{\sum_{i : w \in s_i} H(s_i,w_j)/|s_i|}.$$  

And, the probability $P(w_j|w_i)$ is calculated by the following equation to obtain the word unit entropy from word class sequences. Here, $|c_i|$ is the number of words in word class $c_i$, $H(c_i)$ and $H(c_i,c_j)$ means the occurrence frequency of $c_i$ and $c_i,c_j$, respectively.

$$P(w_j|w_i) = \frac{H(w_i,w_j)}{H(w_i)}$$

$$= \frac{\sum_{i : w \in s_i} \sum_{j : w \in s_j} H(c_i,c_j)/(|c_i| \times |c_j|)}{\sum_{i : w \in s_i} H(c_i)/|c_i|}.$$  

In the case of quasi-trigram, the probability $P(w_k|w_i,w_j)$ is calculated from the occurrence frequencies of $H(s_i,c_i,c_k)$ and $H(s_i,c_j)$ as following:

$$P(w_k|w_i,w_j) = \frac{H(w_i,w_j,w_k)}{H(w_i,w_j)}$$

$$= \frac{\sum_{i : w \in s_i} \sum_{j : w \in s_j} \sum_{k : w \in s_k} H(s_i,c_i,c_k)/(|s_i| \times |c_i| \times |c_k|)}{\sum_{i : w \in s_i} \sum_{j : w \in s_j} H(s_i,c_j)/(|s_i| \times |c_j|)}.$$  

2.2 Test set perplexity

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 “$w_1,w_2\cdots w_n \cdots$”, the test set perplexity is defined as follows:

$$\text{Perplexity} = \exp \left( \frac{1}{n} \sum_{i=1}^{n} \log P(w_i) \right)$$
• General formula

\[ F_T(p) = \left( \frac{1}{P(w_1 | \emptyset)} \times \frac{1}{P(w_2 | \emptyset w_1)} \times \frac{1}{P(w_3 | \emptyset w_1 w_2)} \right. \]
\[ \times \left. \cdots \times \frac{1}{P(w_n | \emptyset w_1 w_2 \cdots w_{n-1})} \times \frac{1}{P(\emptyset | \emptyset w_1 w_2 \cdots w_n)} \right)^{1/(s+1)} \]

• Bigram

\[ F_T(p) = \left( \frac{1}{P(w_1 | \emptyset)} \times \frac{1}{P(w_2 | w_1)} \right. \]
\[ \times \left. \cdots \times \frac{1}{P(w_n | w_{n-1})} \times \frac{1}{P(\emptyset | w_n)} \right)^{1/(s+1)} \]

• Trigram

\[ F_T(p) = \left( \frac{1}{P(w_1 | \emptyset)} \times \frac{1}{P(w_2 | \emptyset w_1)} \right. \]
\[ \times \left. \cdots \times \frac{1}{P(\emptyset | w_{n-1} w_n)} \times \frac{1}{P(\emptyset | w_n)} \right)^{1/(s+1)} \]

In the case of quasi-trigram or wordclass-based bigram, we calculate the test set perplexity using probabilities \( P(w_k | w_l w_j) \) and \( P(w_l | w_j) \) 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 special 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 \( w_1 \) \( w_2 \cdots w_{t-1} \) is \( c_t \), then the test set perplexity is the following:

\[ F(p) = (c_1 c_2 \cdots c_n)^{1/n}. \]

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

3. Language Models

3.1 Task

The experiments were implemented in the “UNIX-QA” task with the vocabulary size of 521 words and performed 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. 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 postpositions
C: case grammar loosing the restriction in B
Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and Quasi/Simplified-Trigram

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

<table>
<thead>
<tr>
<th></th>
<th>sent1</th>
<th>sent2</th>
<th>sent_mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>with duplicated</td>
<td>2048</td>
<td>3000</td>
<td>5048</td>
</tr>
<tr>
<td>sentences</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>without duplicated</td>
<td>1678</td>
<td>2193</td>
<td>3624</td>
</tr>
<tr>
<td>sentence sequences</td>
<td>6.5 x 10^9</td>
<td>2.4 x 10^9</td>
<td>6.7 x 10^9</td>
</tr>
<tr>
<td>generated from</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wordclass sequences</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. The word unit perplexities for bigram

(a) using no flooring method

<table>
<thead>
<tr>
<th></th>
<th>test set perplexity</th>
<th>test set perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>perplexity1</td>
<td>perplexity2</td>
</tr>
<tr>
<td>word</td>
<td>0/1</td>
<td>word 0/1</td>
</tr>
<tr>
<td>class</td>
<td></td>
<td>word</td>
</tr>
<tr>
<td>sent1</td>
<td>7.8</td>
<td>4.7</td>
</tr>
<tr>
<td>sent2</td>
<td>7.1</td>
<td>4.5</td>
</tr>
<tr>
<td>sent_mix</td>
<td>7.6</td>
<td>4.8</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>test set perplexity</th>
<th>test set perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>perplexity1</td>
<td>perplexity2</td>
</tr>
<tr>
<td>word</td>
<td>0/1</td>
<td>word 0/1</td>
</tr>
<tr>
<td>class</td>
<td></td>
<td>word</td>
</tr>
<tr>
<td>sent1</td>
<td>18.3</td>
<td>14.7</td>
</tr>
<tr>
<td>sent2</td>
<td>14.2</td>
<td>13.8</td>
</tr>
<tr>
<td>sent_mix</td>
<td>11.9</td>
<td>12.7</td>
</tr>
</tbody>
</table>

Flooring method is that replaces the frequency of 0 with 0.05. This value was decided by the viewpoint that the test set perplexity and the perplexity 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 restricted CFG (perplexity = 10.0).

The reason, that the word class unit perplexities in the “class” columns are small 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-
Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and Quasi/Simplified-Trigram

Table 3 The word unit perplexities for quasi-trigram

(a) using no flooring method

<table>
<thead>
<tr>
<th></th>
<th>perplexity</th>
<th>test set perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>perplexity1</td>
</tr>
<tr>
<td></td>
<td>word 0/1</td>
<td>class</td>
</tr>
<tr>
<td>sent1</td>
<td>4.1</td>
<td>10.0</td>
</tr>
<tr>
<td>sent2</td>
<td>3.9</td>
<td>9.7</td>
</tr>
<tr>
<td>sent_mix</td>
<td>4.1</td>
<td>10.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>perplexity</th>
<th>test set perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>perplexity1</td>
</tr>
<tr>
<td></td>
<td>word 0/1</td>
<td>class</td>
</tr>
<tr>
<td>sent1</td>
<td>17.3</td>
<td>-</td>
</tr>
<tr>
<td>sent2</td>
<td>12.7</td>
<td>-</td>
</tr>
<tr>
<td>sent_mix</td>
<td>8.9</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4 The word unit perplexities for simplified-trigram

<table>
<thead>
<tr>
<th></th>
<th>test set perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>perplexity1</td>
</tr>
<tr>
<td></td>
<td>word 0/1</td>
</tr>
<tr>
<td>sent1</td>
<td>10.2</td>
</tr>
<tr>
<td>sent2</td>
<td>11.7</td>
</tr>
<tr>
<td>sent_mix</td>
<td>11.0</td>
</tr>
</tbody>
</table>

culated the perplexities for each sentence sets. Though the number of word class­trio is $16 \times 268 \times 268 = 1149184$, 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 “$\square$” 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 $P(w_k \mid w_i w_j)$ for the bigram is the product of the probabilities of normal bigram $P(w_k \mid w_j)$ and bigram $P(w_k \mid w_i)$. Of course, these probability should be normalized, that is,

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

$$\alpha = \frac{1}{\sum \limits_{k} P(w_k \mid w_j) \cdot P(w_k \mid w_i)}.$$
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 Context-Free Grammar

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 probabilities. 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 analyzed and each rule used to generate the sentences also be counted. The learning 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

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

which means the probabilities that the non-terminal symbol \(\alpha\) at the left side will be replaced with the right side patterns \(\beta\). When the SCFG is in the Chomsky normal form, \(\alpha\) is a non-terminal and \(\beta\) 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 analyzed 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

\[
Prob\ (D(i,j)) = \prod\ Prob\ (\alpha \rightarrow \beta).
\]

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

\[
C(i,\alpha \rightarrow \beta) = \frac{\sum_j \text{Prob (D(i,j)) x num (\alpha \rightarrow \beta)}}{\sum_k \text{Prob (D(i,k))}},
\]

After normalization of \(C(i,\alpha \rightarrow \beta)\)

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

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>
<thead>
<tr>
<th>Table 5 The word unit perplexities of SCFG (beam width=45)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample data</td>
</tr>
<tr>
<td>word</td>
</tr>
<tr>
<td>sent_mix</td>
</tr>
</tbody>
</table>
4. RECOGNITION EXPERIMENTS

4.1 Recognition System and Speech Materials

Figure 2 illustrates the recognition system (SPOJUS-SYNO). At first, this system makes word-based HMMs automatically by concatenating syllable-based (trained) HMMs. Japanese syllables consist of about 110 syllables each of which is composed of a consonant and a vowel (CV), a syllabic nasal (N), a vowel (V), or a consonant, a semivowel 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 calculated by using 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-right 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 right-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 sentences 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

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. 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)

<table>
<thead>
<tr>
<th>perplexity</th>
<th>10.0</th>
<th>15.9</th>
<th>19.3</th>
<th>25.4</th>
<th>50.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>speaker</td>
<td>sent.rec.%</td>
<td>sent.rec.%</td>
<td>sent.rec.%</td>
<td>sent.rec.%</td>
<td>sent.rec.%</td>
</tr>
<tr>
<td>est.</td>
<td>exp.</td>
<td>est.</td>
<td>exp.</td>
<td>est.</td>
<td>exp.</td>
</tr>
<tr>
<td>TI</td>
<td>76.5</td>
<td>78.7</td>
<td>66.1</td>
<td>66.0</td>
<td>61.4</td>
</tr>
<tr>
<td>MA</td>
<td>54.6</td>
<td>79.6</td>
<td>40.8</td>
<td>46.9</td>
<td>35.2</td>
</tr>
<tr>
<td>KG</td>
<td>67.2</td>
<td>71.7</td>
<td>55.1</td>
<td>58.7</td>
<td>49.8</td>
</tr>
<tr>
<td>SE</td>
<td>77.7</td>
<td>60.0</td>
<td>70.5</td>
<td>61.1</td>
<td>51.1</td>
</tr>
<tr>
<td>SN</td>
<td>60.9</td>
<td>50.0</td>
<td>46.1</td>
<td>25.0</td>
<td>40.8</td>
</tr>
<tr>
<td>average</td>
<td>67.5</td>
<td>68.7</td>
<td>55.9</td>
<td>51.1</td>
<td>50.9</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>speaker</th>
<th>input word</th>
<th>% correct</th>
<th>subst.</th>
<th>ins.</th>
<th>del.</th>
<th>sent. rec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>341</td>
<td>91.2%</td>
<td>90.3%</td>
<td>6.7%</td>
<td>0.9%</td>
<td>2.1%</td>
</tr>
<tr>
<td>MA</td>
<td>372</td>
<td>84.4</td>
<td>82.8</td>
<td>12.9</td>
<td>1.6</td>
<td>2.7</td>
</tr>
<tr>
<td>HU</td>
<td>328</td>
<td>87.5</td>
<td>85.1</td>
<td>10.7</td>
<td>2.4</td>
<td>1.8</td>
</tr>
<tr>
<td>KG</td>
<td>334</td>
<td>88.3</td>
<td>86.5</td>
<td>9.6</td>
<td>1.8</td>
<td>2.1</td>
</tr>
<tr>
<td>SE</td>
<td>319</td>
<td>86.2</td>
<td>83.4</td>
<td>12.5</td>
<td>2.8</td>
<td>1.3</td>
</tr>
<tr>
<td>SN</td>
<td>357</td>
<td>73.4</td>
<td>64.4</td>
<td>22.7</td>
<td>9.0</td>
<td>3.9</td>
</tr>
<tr>
<td>average</td>
<td>342</td>
<td>85.2</td>
<td>82.1</td>
<td>12.5</td>
<td>3.1</td>
<td>2.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>speaker</th>
<th>input word</th>
<th>% correct</th>
<th>subst.</th>
<th>ins.</th>
<th>del.</th>
<th>sent. rec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>341</td>
<td>85.0%</td>
<td>94.7%</td>
<td>3.8%</td>
<td>0.3%</td>
<td>1.2%</td>
</tr>
<tr>
<td>MA</td>
<td>372</td>
<td>87.1</td>
<td>85.5</td>
<td>10.8</td>
<td>1.6</td>
<td>2.2</td>
</tr>
<tr>
<td>HU</td>
<td>328</td>
<td>87.5</td>
<td>86.3</td>
<td>10.4</td>
<td>1.2</td>
<td>2.1</td>
</tr>
<tr>
<td>KG</td>
<td>334</td>
<td>87.7</td>
<td>85.6</td>
<td>9.9</td>
<td>2.1</td>
<td>2.4</td>
</tr>
<tr>
<td>SE</td>
<td>319</td>
<td>88.1</td>
<td>87.2</td>
<td>10.7</td>
<td>0.9</td>
<td>1.3</td>
</tr>
<tr>
<td>SN</td>
<td>357</td>
<td>73.7</td>
<td>68.1</td>
<td>23.8</td>
<td>5.6</td>
<td>2.5</td>
</tr>
<tr>
<td>average</td>
<td>342</td>
<td>86.5</td>
<td>84.6</td>
<td>11.7</td>
<td>2.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Comparison of Language Models by Stochastic Context-Free Grammar, Bigram and Quasi/Simplified-Trigram

Table 9 Recognition results using simplified-trigram for sent_mix (beam width=100) (test set perplexity2 = 14.9)

<table>
<thead>
<tr>
<th>speaker</th>
<th>input</th>
<th>word acc.</th>
<th>% correct</th>
<th>subst.</th>
<th>ins.</th>
<th>del.</th>
<th>sent. rec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>341</td>
<td>94.7%</td>
<td>93.6%</td>
<td>4.4%</td>
<td>1.2%</td>
<td>0.9%</td>
<td>70.2%</td>
</tr>
<tr>
<td>MA</td>
<td>372</td>
<td>88.4</td>
<td>87.8</td>
<td>9.7</td>
<td>0.8</td>
<td>1.9</td>
<td>61.2</td>
</tr>
<tr>
<td>HU</td>
<td>328</td>
<td>92.1</td>
<td>90.6</td>
<td>6.1</td>
<td>1.5</td>
<td>1.8</td>
<td>65.2</td>
</tr>
<tr>
<td>KO</td>
<td>334</td>
<td>91.0</td>
<td>90.1</td>
<td>6.9</td>
<td>0.9</td>
<td>2.1</td>
<td>63.0</td>
</tr>
<tr>
<td>SE</td>
<td>319</td>
<td>91.5</td>
<td>90.6</td>
<td>7.8</td>
<td>0.9</td>
<td>0.6</td>
<td>57.8</td>
</tr>
<tr>
<td>SN</td>
<td>357</td>
<td>72.0</td>
<td>59.1</td>
<td>25.2</td>
<td>12.9</td>
<td>2.8</td>
<td>27.1</td>
</tr>
<tr>
<td>average</td>
<td>342</td>
<td>88.3</td>
<td>85.3</td>
<td>10.0</td>
<td>3.0</td>
<td>1.7</td>
<td>57.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>speaker</th>
<th>TI</th>
<th>MA</th>
<th>HU</th>
<th>KO</th>
<th>SE</th>
<th>SN</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentence recognition rate (%)</td>
<td>78.7</td>
<td>78.6</td>
<td>73.3</td>
<td>69.6</td>
<td>62.2</td>
<td>54.2</td>
<td>69.5</td>
</tr>
</tbody>
</table>

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 corresponding 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 CFG (in the case of the grammar A: perplexity=10.0) using SUN-SPARK station-1
Seiichi Nakagawa, Isao Murass, and Min Zhou

(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 $16 \times 521 \times 521$). For these things, the processing time of the quasi-trigram model becomes to be later.

Second, the perplexity 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

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

Advanced Man-Machine Interface Through Spoken Language (1988)


