AN EVALUATION TO DETECT AND CORRECT ERRONEOUS CHARACTERS WRONGLY SUBSTITUTED, DELETED AND INSERTED IN JAPANESE AND ENGLISH SENTENCES USING MARKOV MODELS

Tetsuo ARAKI† Satoru IKEHARA†† Nobuyuki TSUKAHARA† Yasunori KOMATSU†

† Faculty of Engineering, Fukui University Fukui, 910 JAPAN
†† NTT Communication Science Laboratories 1-2356 Take Yokosuka-Shi 238-03 Japan

ABSTRACT

In optical character recognition and continuous speech recognition of a natural language, it has been difficult to detect error characters which are wrongly deleted and inserted. In order to judge three types of the errors, which are characters wrongly substituted, deleted or inserted in a Japanese "bunsetsu" and an English word, and to correct these errors, this paper proposes new methods using m-th order Markov chain model for Japanese "kanji-kana" characters and English alphabets, assuming that Markov probability of a correct chain of syllables or "kanji-kana" characters is greater than that of erroneous chains.

From the results of the experiments, it is concluded that the methods is useful for detecting as well as correcting these errors in Japanese "bunsetsu" and English words.

Key words: Markov model, error detection, error correction, bunsetsu, substitution, deletion, insertion

1 Introduction

In order to improve the man-machine interface with computers, the development of input devices such as optical character readers (OCR) or speech recognition devices are expected. However, it is not easy to input Japanese sentences by these devices, because they are written by many kinds of characters, especially thousands of "kanji" characters. The sentences input through an OCR or a speech recognition device usually contain erroneous character strings.

The techniques of natural language processing are expected to find and correct these errors. However, since current technologies of natural language analysis have been developed for correct sentences, they cannot directly be applied to these problems. Up to now, statistical approaches have been made to this problem.

Markov models are considered to be one of machine learning models, similar to neural networks and fuzzy models. They have been applied to character chains of natural languages (e.g.,English)[1],[2], and to phoneme chains in continuous speech recognition[3],[4]. 2nd-order Markov model in "bunsetsu" is known to be useful to correct errors in "kanji-kana" "bunsetsu" [6],to choose a correct syllable chain from Japanese syllable "bunsetsu" candidates [7], and to reduce the ambiguities in translation processing of non-segmented "kana" sentences into "kanji-kana" sentences [8].

The erroneous characters can be classified into three types. The first is wrongly recognized characters instead of correct characters. The second and the third are wrongly inserted and deleted (skipped) characters respectively. Markov chain models above mentioned were restricted to find and correct the first type of errors[5],[6]. No method has been proposed for correcting errors of the second and the third types. The reason might be considered to be the difficulties of finding the error location and distinguishing between deletion and insertion errors.

On the other hand, contextual algorithm utilizing n-gram letter statistics (e.g.[9]) and a dictionary look-up algorithm[10] have been discussed to detect and correct erroneous characters in English sentences, which is segmented into words.

This paper proposes new methods, which are able to be applied to a non-segmented chains of characters, to judge three types of the errors, which are characters wrongly substituted, deleted and inserted in a Japanese "bunsetsu", and to correct these errors in Japanese "kanji-kana" chains using m-th order Markov chain model. The methods are based on the idea about the relation between the types of errors and the length of a chain in which the values of Markov joint probability remain small. Furthermore, this method is applied to detect and correct errors in segmented English words.

Experiments were conducted for the case of 2nd-order and 3rd-order Markov model, and they were applied to Japanese and English newspaper articles. "Relevance Factor" P and "Recall Factor" R for erroneous characters detected and corrected by this method were experimentally evaluated using statistical data for 70 issues of a daily Japanese newspaper and 5 issues of a daily English newspaper.

2 Basic Definitions and the Method of Error Detection and Error Correction using 2nd-Order Markov Model

2.1 Basic Definitions

In this paper, two types of natural language's sentences are discussed. One is a Japanese sentence, which is non-segmented sentence and the other is an English sentence, which is segmented into words.

A Japanese sentence can be separated into syntactic units called "bunsetsu", where a "bunsetsu" is composed of one "independent word" and a sequence of n (greater than equal to 0) "dependent words".

A "bunsetsu" is a chain of Japanese "kanji-kana" characters or an English word is a chain of alphabets, and are represented by $\gamma = s_1 s_2 \cdots s_n$, where s_i is a "kanji-kana" character or an alphabet. In particular, a chain, γ , is called a "J-bunsetsu" when all of its elements are "kanji-kana" characters, and is called a "E-word" when all of its elements are English alphabets. The set of corrçct Japanese "bunsetsu" or English words is represented by Γ_C.

Three types of erroneous "J-bunsetsu" or "E-word" are defined as follows:

First, a chain $\alpha = \bar{s_1}\bar{s_2} \cdots \bar{s_{i-1}}\bar{s_i} \cdots \bar{s_m}$ is called a "(i,k)-Erroneous J-bunsetsu or E-word Wrongly Substituted " ($(i,k)-$EWS) if a subchain $\beta = t_1 t_2 \cdots t_k$ is wrongly substituted at the location i of α, that is $\exists \gamma \in \Gamma_C, \quad \gamma = \alpha^{(i)} \| \beta$. Here $\alpha^{(i)} \| \beta$ denotes substitution of a subchain β at the location i in a chain α , that is, $\alpha^{(i)} \| \beta \equiv \bar{s_1}\bar{s_2} \cdots \bar{s_{i-1}} t_1 t_2 \cdots t_k \bar{s_{i+k}} \cdots \bar{s_m}$, and $t_1 \leftarrow \bar{s_i}, \cdots, t_k \leftarrow \bar{s_{i+k-1}}$.

Next, a chain $\alpha = \check{s_1}\check{s_2} \cdots \check{s_{i-1}}\check{s_i} \cdots \check{s_m}$ is called a "(i,k)-Erroneous J-bunsetsu or E-word Wrongly Deleted" ($(i,k)-$EWD) if a subchain $\beta = t_1 t_2 \cdots t_k$ is wrongly deleted at the location i of α, that is $\exists \gamma \in \Gamma_C, \quad \gamma = \alpha^{(i)} \ll \beta$. Here $\alpha^{(i)} \ll \beta$ denotes insertion of a subchain β at the location i in a chain α , that is, $\alpha^{(i)} \ll \beta \equiv \check{s_1}\check{s_2} \cdots \check{s_{i-1}} t_1 t_2 \cdots t_k \check{s_i} \cdots \check{s_m}$.

Finally, a chain $\alpha = \hat{s_1} \cdots \hat{s_{i-1}}\hat{s_i} \cdots \hat{s_{i+k-1}}\ \hat{s_{i+k}} \cdots \hat{s_m}$ is also called "(i,k)-Erroneous J-bunsetsu or E-word Wrongly Inserted" ($(i,k)-$EWI) if a subchain $\beta = t_1 t_2 \cdots t_k$ is wrongly inserted at the location i of α, that is $\exists \gamma \in \Gamma_C, \quad \gamma = \alpha^{(i)} \gg \beta$. Here $\alpha^{(i)} \gg \beta$ denotes deletion of a subchain β at the location i in a chain α , that is, $\alpha^{(i)} \gg \beta \equiv \hat{s_1}\hat{s_2} \cdots \hat{s_{i-1}}\hat{s_{i+k}} \cdots \hat{s_m}$, and $t_1 = \hat{s_i}, \cdots, t_k = \hat{s_{i+k-1}}$.

The set of (i,k)-EWS, (i,k)-EWD and (i,k)-EWI are represented by $\Gamma_S^{(k)}$, $\Gamma_D^{(k)}$ and $\Gamma_I^{(k)}$ respectively. In this paper, all inputs "bunsetsu" or all inputs words to computers are assumed to belong to one of Γ_C , $\Gamma_S^{(k)}$, $\Gamma_D^{(k)}$ and $\Gamma_I^{(k)}$.

Next, the meaning of detecting and correcting errors are defined in the following. The words, "error detection problem", means the problem how to detect the location i of error in α, and "error correction problem" means the problem how to replace an erroneous "J-bunsetsu" or an "E-word" α by a correct "bunsetsu" or an English word γ, where $\alpha \in \Gamma_S^{(k)}, \alpha \in \Gamma_D^{(k)}, \, or \, \alpha \in \Gamma_I^{(k)}$, and $\gamma \in \Gamma_C$.

"Relevance Factor" $P^{(D)}$ and "Recall Factor" $R^{(D)}$ for the "error detection problem" is defined as follows:

(1): $P^{(D)} \equiv$ (the number of "J-bunsetsu" or "E-word" that the location i and length k of error in $\Gamma_S^{(k)}$, $\Gamma_D^{(k)}$ or $\Gamma_I^{(k)}$ is correctly detected) / (the total number of "J-bunsetsu" or "E-word" detected as erroneous "J-bunsetsu" or "E-word").

(2): $R^{(D)} \equiv$ (the number of "J-bunsetsu" or "E-word" that the location i and length k of error in $\Gamma_S^{(k)}$, $\Gamma_D^{(k)}$ or $\Gamma_I^{(k)}$ is correctly detected) / (the number of all "J-bunsetsu" or "E-word" in the set $\Gamma_S^{(k)}$, $\Gamma_D^{(k)}$ or $\Gamma_I^{(k)}$ prepared in advance).

"Relevance factor" $P^{(C)}$ and "Recall factor" $R^{(C)}$ for the "error correction problem" is also similarly defined. Here $P_I^{(D)}$ denotes the "Relevance Factor" for the "error detection problem" of $\Gamma_I^{(k)}$, and $R_D^{(C)}$ denotes the "Recall Factor" for the "error correction problem" of $\Gamma_D^{(k)}$ respectively.

2.2 The Method of Error Detection using 2nd-Order Markov Model

We introduce the following assumption according to the experiences.

Assumption Each Markov probability for erroneous chains of "kanji-kana" characters or English alphabets is small compared to that of correct chains.

According to this assumption, the procedure of detecting the location i and the length k of error chains are defined as follows:

Procedure 1 (Method of detecting the location and the length of chain wrongly substituted in $\Gamma_S^{(k)}$ and substituted or inserted in $\Gamma_I^{(k)}$)
Find the subchain of length k which satisfy the following conditions. This chain is judged to be wrongly inserted at the location i.

(1) $P(X_h \mid X_{h-m} \cdots X_{h-1}) > T$, for $h = i - 1$ or $h = i + k + m$ and

(2) $P(X_j \mid X_{j-m} \cdots X_{j-1}) < T$, for $\forall j$ such that $i \leq j \leq i + k + m - 1$,

where $P(X_j \mid X_{j-m} \cdots X_{j-1})$ is m-th order Markov chain probability which denotes probability of occurrence of successive character X_j when string X_{j-m} $\cdots X_{j-1}$ has occurred, and X_u denotes a space symbol if $u < 0$. And T denotes a critical value of m-th order Markov probability used for detecting errors.

This procedure detects that k characters are wrongly substituted or inserted at the location i, if m-th order Markov probability for chain remain smaller value than critical value T just $(k+m)$ times from the location i to $i+k+m-1$.

For an example, the change of the value of 2nd-order Markov probability for each character of the erroneous chain $\Gamma_S^{(2)}$ or $\Gamma_I^{(2)}$ is shown in Fig. 1. In this example, two characters are wrongly substituted or inserted. According to the previous assumption, 2nd-order Markov probability for erroneous chain remain smaller value than critical value T just four times.

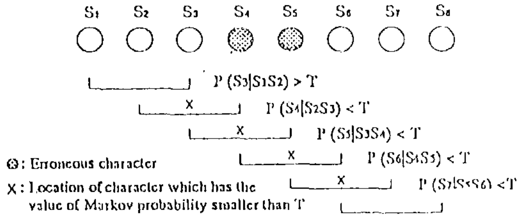


Fig.1. Change of the value of 2nd-order Markov probabilities for each character of the erroneous string including wrongly substituted or inserted characters

Procedure 2 (Method of detecting the location of chain wrongly deleted in $\Gamma_D^{(k)}$)
Find the subchain of length k which satisfy the following conditions. This chain is judged to be wrongly deleted at the location i.

(1) $P(X_h \mid X_{h-m} \cdots X_{h-1}) > T$, for $h = i - 1$ or $h = i + k + m$ and

(2) $P(X_j \mid X_{j-m} \cdots X_{j-1}) < T$, for $\forall j$ such that $i \leq j \leq i + m - 1$,

where T denotes a critical value of m-th order Markov probability used for detecting errors.

If m-th order Markov probabilities for chain remain smaller than the critical value T just m times from the location i to $i + m - 1$, it is judged that some characters are wrongly deleted at the location i. However note that length k of characters wrongly deleted at the location i, can not be determined by this procedure, the length k is determined by the procedure 4 shown in Sec. 2.3.

Table 1 shows that the relation of times that Markov probabilities remain smaller than T in the cases of 1st- and 2nd-order Markov models. Erroneous chains can be classified into the following two cases: one is a case of the characters wrongly substituted or inserted, the other is a class of the characters wrongly deleted.

Table 1[1] The number of times that Markov probability of the erroneous chains remain a smaller than T

type	1st-order Markov	2nd-order Markov
$\Gamma_S^{(1)}$	two times	three times
$\Gamma_S^{(2)}$	three times	four times
$\Gamma_S^{(k)}$	$(k+1)$ times	$(k+2)$ times
$\Gamma_D^{(1)}$	one times	two times
$\Gamma_D^{(2)}$	one times	two times
$\Gamma_D^{(k)}$	one times	two times
$\Gamma_I^{(1)}$	two times	three times
$\Gamma_I^{(2)}$	three times	four times
$\Gamma_I^{(k)}$	$(k+1)$ times	$(k+2)$ times

[1] In case to detect errors in $\Gamma_I^{(2)}$ using 2nd-order Markov model, it is able to presumed that a subchain β of length 2 is wrongly inserted at the location i of erroneous chain α, if 2nd-order Markov probability for erroneous chain α remain smaller than T just four times from location i.

However, this method can not distinguish the erroneous characters wrongly substituted, from the characters wrongly inserted in the former case, and can not determine the length k for the type of $\Gamma_D^{(k)}$, because the Markov probability of any erroneous chains in $\Gamma_D^{(k)}$ remains small value just the same times for length k. These problems can be solved by the procedure 3 and 4 shown in Sec.2.3.

In this paper, the effect to detect errors for cases of length $k = 1, 2$ is evaluated.

2.3 The Method of Error Correction using 2nd-Order Markov Model

The procedure of replacing erroneous chains by correct chains using Markov model is presented as follows:

Procedure 3 (Method of correcting the chains in $\Gamma_S^{(k)}$ or $\Gamma_I^{(k)}$)

"bunsetsu"

or words $\alpha = \bar{s}_1\bar{s}_2\cdots \bar{s}_{i-1}\bar{s}_i\cdots \bar{s}_{i+k-1}\bar{s}_{i+k}$ or $\alpha = \hat{s}_1\hat{s}_2\cdots \hat{s}_{i-1}\hat{s}_i\cdots \hat{s}_{i+k-1}\hat{s}_{i+k}\cdots \hat{s}_m$ denotes a " (i, k)-EWS" and a "(i, k)-EWI" and a subchain $\beta = t_1t_2\cdots t_k$ is assumed to be wrongly substituted or inserted at the location i of α respectively. Then the erroneous chain α can be replaced by the following correct chain γ in Γ_C if condition (1) is satisfied.

$\gamma{=}\alpha^{(i)} \| \beta \equiv \bar{s}_1\bar{s}_2\cdots \bar{s}_{i-1}t_1t_2\cdot t_k\bar{s}_{i+k}\cdots \bar{s}_m$, and $t_1 \leftarrow \bar{s}_i, \cdots, t_k \leftarrow \bar{s}_{i+k-1}$ or $\gamma{=}\alpha^{(i)} \gg \beta \equiv \hat{s}_1\hat{s}_2\cdots \hat{s}_{i-1}\hat{s}_{i+k}\cdots \hat{s}_m$, and $t_1 = \hat{s}_i, \cdots, t_k = \hat{s}_{i+k-1}$

(1) $P(X_j \mid X_{j-m}\cdots X_{j-1}) > T$ for $\forall j$ such that $i + k \leq j \leq i + k + m - 1$.

By comparing Markov probability for correct chains in two cases above, choose a correct chain which has the great Markov probability.

∎

Procedure 4 (Method of correcting the erroneous chains in $\Gamma_D^{(k)}$)

A chain $\alpha = \check{s}_1\check{s}_2\cdots \check{s}_{i-1}\check{s}_i\cdots \check{s}_m$ denotes a "(i, k)-EWD" and a subchain $\beta = t_1t_2\cdots t_k$ is assumed to be wrongly deleted at the location i of α. Then the erroneous chain α can be replaced by the following correct chain γ in Γ_C if condition (1) is satisfied.

$\gamma = \alpha^{(i)} \ll \beta$
$\equiv \quad \bar{s}_1\bar{s}_2\cdots \bar{s}_{i-1}t_1t_2\cdots t_k\bar{s}_i\cdots \bar{s}_m.$

(1) $P(X_j \mid X_{j-m}\cdots X_{j-1}) > T$, for $\forall j$ such that $i + k \leq j \leq i + k + m - 1$. ∎

An example of correcting the erroneous chain, two characters of which are wrongly substituted ($\Gamma_S^{(2)}$), is shown in Fig. 2. If Markov probabilities do not remain smaller than critical value T, then it is judged that these erroneous chains have been corrected.

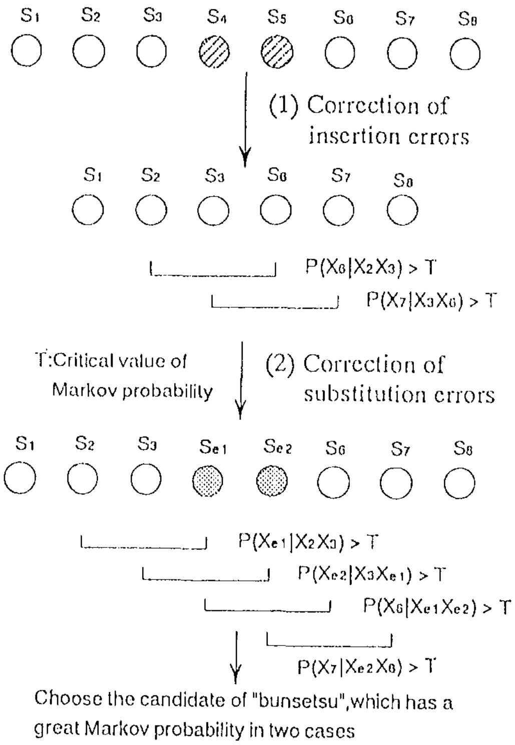

S1 S2 S3 S4 S5 S6 S7 S8
(1) Correction of insertion errors
S1 S2 S3 S6 S7 S8
P(X6|X2X3) > T
P(X7|X3X6) > T
T:Critical value of Markov probability
(2) Correction of substitution errors
S1 S2 S3 Se1 Se2 S6 S7 S8
P(Xe1|X2X3) > T
P(Xe2|X3Xe1) > T
P(X6|Xe1Xe2) > T
P(X7|Xe2X6) > T
Choose the candidate of "bunsetsu",which has a great Markov probability in two cases

Fig.2 Procedure for correcting an erroneous string using error detection

3 Experimental Results

3.1 Experimental Conditions

1. The number of "bunsetsu" for 70 issues of a daily Japanese newspaper: 283,963 "bunsetsu"
2. The number of words for 5 issues of a daily English newspaper: 155,459 words
3. Type of errors and the number of "bunsetsu":
   800 "bunsetsu" are prepared for each of $\Gamma_S^{(1)}, \Gamma_S^{(2)}, \Gamma_D^{(1)}, \Gamma_D^{(2)}, \Gamma_I^{(1)}$, and $\Gamma_I^{(2)}$.
   (a) The average length of "bunsetsu" composed of "kanji-kana" character chains: 6 characters
   (b) The average length of alphabets composed of correct English words chains : 7 characters
4. Markov model of Japanese "kanji-kana" characters : 2nd-order Markov Model
5. Markov models of English alphabets: 2nd- and 3rd-order Markov models

3.2 Experimental Results and Discussion

The accuracy of error detection and error correction depends on the critical value T of Markov probabilities. "Relevance Factor" P and "Recall Factor" R for each method were obtained by changing the value of T.

[1] The Relation between P and R of Detecting Erroneous Chain Using Detection Procedure

The relation between P and R for the location of erroneous "kanji-kana" chains detected in $\Gamma_S^{(1)}$, $\Gamma_S^{(2)}$, $\Gamma_D^{(1)}$, $\Gamma_D^{(2)}$, $\Gamma_I^{(1)}$, and $\Gamma_I^{(2)}$ using *Procedure* 1 and 2, are shown in Fig. 3, and those for erroneous alphabets chains are shown in Fig. 4.

From these figures, the following results are obtained :

1. The maximum value of P and R of detecting erroneous characters wrongly inserted or substituted, is greater than that of erroneous characters wrongly deleted.

(a) In the case of "J-bunsetsu" :

$$P_I^{(D)} = 97-99\%, \quad R_I^{(D)} = 97-99\%$$
$$P_D^{(D)} = 100\%, \quad R_D^{(D)} = 57-58\%$$
$$P_S^{(D)} = 88-94\%, \quad R_S^{(D)} = 88-93\%$$

(b) In the case of "E-word":

$$P_I^{(D)} = 38-49\%, \quad R_I^{(D)} = 38-39\%$$
$$P_D^{(D)} = 94-95\%, \quad R_D^{(D)} = 16-19\%$$
$$P_S^{(D)} = 42-58\%, \quad R_S^{(D)} = 39-42\%$$

2. Compared with these maximal values, it is shown that the maximum value of product of P and R for "kanji-kana" "bunsetsu" is 35%−60% greater than that of English words.

[2] The Relation between P and R of Chains Corrected Using Correction Procedure

The relation between P and R of "J-bunsetsu" corrected using *Procedure* 3 and 4 for $\Gamma_S^{(1)}, \Gamma_S^{(2)}, \Gamma_D^{(1)}, \Gamma_D^{(2)}, \Gamma_I^{(1)}$, $\Gamma_I^{(2)}$ of "J-bunsetsu" are shown in Fig. 5.

From this figure, the following results are obtained :

The maximum value of P and R of correcting erroneous characters wrongly inserted or substituted, using 2nd-order Markov model, is greater than that of erroneous characters wrongly deleted.

$$P_I^{(C)} = 92-98\%, \quad R_I^{(C)} = 91-97\%$$
$$P_D^{(C)} = 79-86\%, \quad R_D^{(C)} = 46-49\%$$
$$P_S^{(C)} = 69-94\%, \quad R_S^{(C)} = 62-88\%$$

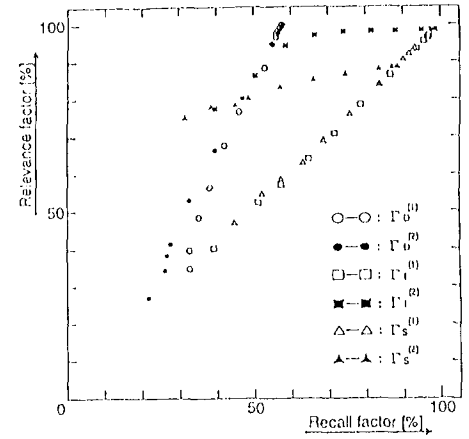


Fig.3. Experimental results for detecting a location of an erroneous "kanji-kana" string using the error detection procedure

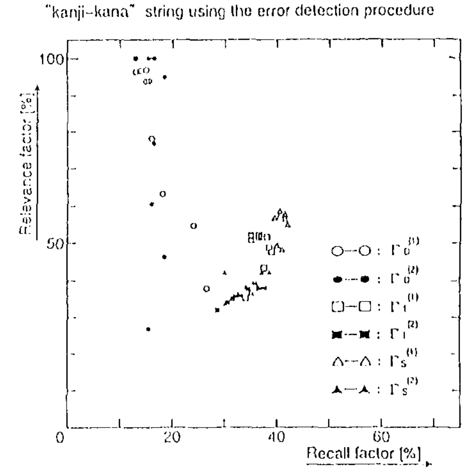


Fig.4. Experiental results for detecting a location of an erroneous English words using the error detection procedure

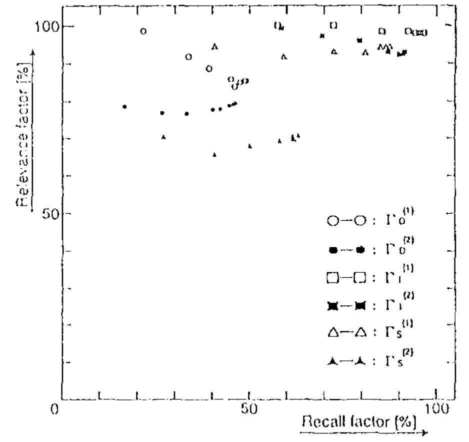


Fig.5. Experimental results for correcting an erroneous "kanji-kana" string using error correction procedure

[3] The Combinatorial Effect to Correct Erroneous English Words Using the Spell Checker and the Correction Procedure by Markov Model

The experimental results of detecting errors in English words using Ispell (Interactive Spell checker) is shown in Table 2. From the results, it is seen that Ispell can almost perfectly detect erroneous words in Γ_I, Γ_D and Γ_S using dictionary, but it cannot perfectly correct erroneous words, because it can output the correct candidates for erroneous words in $\Gamma_I^{(1)}$, $\Gamma_D^{(1)}$, $\Gamma_S^{(1)}$, but can not output the correct candidates for erroneous words in $\Gamma_I^{(2)}$, $\Gamma_D^{(2)}$, $\Gamma_S^{(2)}$. It is necessary to detect the location of erroneous alphabets in words to detect all these errors. However, it should be noted that Ispell can not detect the location of erroneous alphabets in words.

In order to detect and correct erroneous "E-word" more effectively, the method to combine Ispell and the *procedure* (in sec. 2.3) using Markov model is expected. The combinatorial method is denoted in the following way: (1) At first, erroneous "E-words" are detected by Ispell, but the locations of erroneous alphabets in words can not be detected by it. (2) Next decide the correct candidates words by *procedure* 3 and 4. (3) Finally, Ispell again checks if these candidates are correct words. The experimental results using this method is shown in Fig. 6(2nd-order) and in Fig. 7(3rd-order).. From the results, it is seen that this combinatorial method of Ispell and the procedure by 3rd-order Markov model to very useful to detect and correct all errors in English words.

It takes about 10 milli-seconds and 6 seconds in average to detect and to correct erroneous "bunsetsu" . Examples of "bunsetsu" and the output results of error detection and error correction using Markov model, are shown in Fig. 8.

Table 2 The capability of error detection using Ispell

	Able to detect		Unable to detect
	With correct candidate	Without correct candidate	
$\Gamma_D^{(1)}$	76. 0%	17. 5%	6. 5%
$\Gamma_D^{(2)}$	0%	79. 5%	20. 5%
$\Gamma_I^{(1)}$	82. 0%	18. 0%	0%
$\Gamma_I^{(2)}$	0%	100. 0%	0%
$\Gamma_S^{(1)}$	80. 5%	18. 5%	1. 0%
$\Gamma_S^{(2)}$	4. 0%	96. 0%	0%

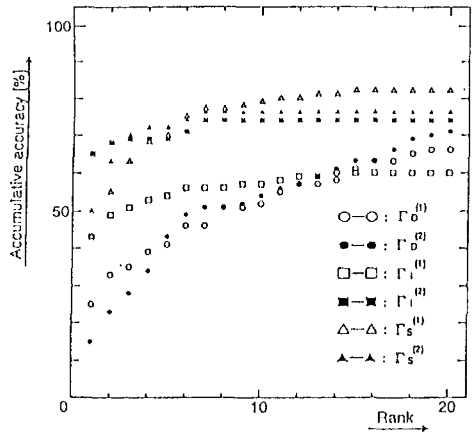


Fig.6. Exprimental result for correcting erroneous English words using Ispell and error correction procedure in case of 2rd-order Markov model

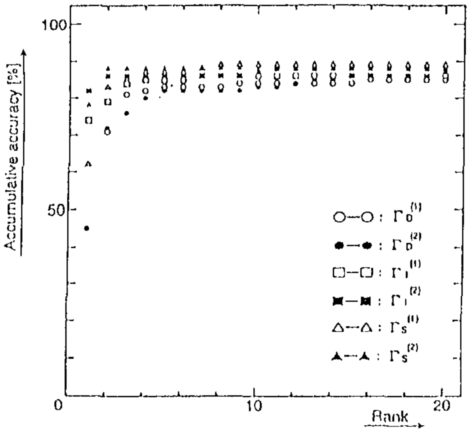
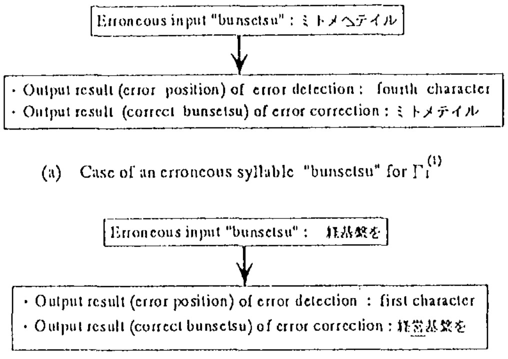


Fig.7. Exprimental result for correcting erroneous English words using Ispell and error correction procedure in case of 3rd-order Markov model



Erroneous input "bunsetsu" : ミトメヘテイル

- Output result (error position) of error detection : fourth character
- Output result (correct bunsetsu) of error correction : ミトメテイル

(a) Case of an erroneous syllable "bunsetsu" for $\Gamma_I^{(1)}$

Erroneous input "bunsetsu" : 拝基紫を

- Output result (error position) of error detection : first character
- Output result (correct bunsetsu) of error correction : 拝啓基紫を

(b) Case of an erroneous "kanji-kana" "bunsetsu" for $\Gamma_D^{(1)}$

Fig.8. Examples of erroneous "bunsetsu" and the results of error detection and error correction

4 Conclusion

This paper proposed the methods to judge three type of errors and correct these errors, which are characters wrongly substituted, inserted and deleted in the Japanese "kanji-kana" chains and English words using m-th order Markov model.

The effects of the methods were experimentally evaluated for the case of 2nd- and 3rd-order Markov chain. From the experimental results, the following conclusions have been obtained:

1. The maximum value of P and R of detecting erroneous characters wrongly inserted or substituted, is greater than that of erroneous characters wrongly deleted.
2. This method is specially useful to detect and correct erroneous characters wrongly inserted and substituted in "kanji-kana" "bunsetsu", but is not so useful to detect and correct errors in English words.
3. The combinatorial method of Ispell and the procedure by 3rd-order Markov model is usefull to detect and correct all errors in English words.

However they are not so useful for detecting and correcting of characters wrongly deleted in "kanji-kana" "bunsetsu". Then, more efficient methods are expected for this type of errors.

References

[1] T.Araki,J.Murakami and S.Ikehara "Effect of Reducing Ambiguity of Recognition Candidates in Japanese Bunsetsu Unit by 2nd-Order Markov Model of Syllables", *Information Processing Society of Japan*, Vol.30, No.4, pp.467-477 (1989)

[2] S.Ikehara and S.Shirai "Japanese Character Error Detection by Word Analysis and Correction Candidate Extraction by 2nd-Order Markov Model ", *Information Processing Society of Japan*, Vol.25, No.2, pp.298-305 (1984)

[3] F.Jelinek "Continuous Speech Recognition by Statistical Methods", *Proc. of the IEEE*, Vol.64, No.4, pp.532-556 (1976)

[4] T. Kurita and T.Aizawa "A Method for Correcting Errors on Japanese Word Input and Its Application to Spoken Word Recognition with Large Vocabulary", *Information Processing Society of Japan*, Vol.25, No.5, pp.831-841 (1984)

[5] J.Murakami,T.Araki and S.Ikehara "The Effect of Trigram Model in Japanese Speech Recognition", *The Institute of Electronics, Information and Communication Engineers*, Vol.J75-D-II, No.1, pp.11-20 (1992)

[6] Y. Ooyama and Y. Miyazaki "Natural Language Processing in a Japanese-text-to-speech System ", *Information Processing Society of Japan*, Vol.27, No.11, pp.1053-1061 (1986)

[7] J.L.Peterson "Computer Programs for Detecting and Correcting Spelling Errors", *Comm., ACM*, Vol. 23, No. 12, pp. 676-687 (1980)

[8] L.R.Rabiner,S.E.Levinson and M.M. Sondai "On the Application of Vector Quantization and Hidden Markov Models to Speaker-independent, Isolated Word Recognition", *Bell System Technical Journal*, Vol.62, No.4, pp.1075-1105 (1983)

[9] E.M.Riseman and A.R. Hanson "A Contextual Postprocessing System for Error Correction Using Binary n-Gram", *IEEE Trans. Comput.*, Vol. C-23, No. 5, pp. 480-493 (1974)

[10] C.E.Shannon "Mathematical Theory of Communication", *Bell System Technical Journal*, Vol.27, pp.379-423, 623-656, October (1948)

[11] C.E.Shannon "Prediction and Entropy of Printed English", *Bell System Technical Journal*, Vol.30, pp.50-64, January (1951)