

綴りの置換誤りの高速訂正法†

田中 榮一^{††} 小橋口 高広^{††} 島村 邦彦^{††}

綴りの誤り訂正法には、大別すると、統計法と辞書法があり、辞書法は訂正率は高いが、訂正に時間がかかることされている。本論文は、置換誤りを訂正するときの辞書法の高速化法を二つ提案し、実験した結果について述べている。OCR で文を入力するとき、文字の読取り誤りにはある偏りがあり、ある文字が他のすべての文字に誤ることは、まずない。タイプの打鍵誤りでも同様である。この偏り情報を利用すると、訂正するために探索する辞書の範囲を、劇的に縮小することができる。探索範囲の縮小は、直接、訂正率の向上と訂正時間の短縮につながる。文字誤りの偏りから、文字が幾つかの類に分割できる場合と、必ずしもそうでない場合があるが、前者に対して高速化法1、後者に対して高速化法2を提案した。英語の6文字語、2,755語について実験したところ、高速化法1の適用できる場合、文字が4類に分かれるなら、1文字誤り訂正では、全辞書法と比べて、訂正率が10%、2文字誤り訂正では、35%向上した。訂正に要する時間は、1/45に短縮された。また、高速化法2が適用できる場合、2文字誤りの訂正では、訂正率が35~40%向上し、訂正に要する時間は1/5に短縮された。

1. ま え が き

綴りの誤り訂正法の研究は、光学的文字読取機(OCR)の読取り誤り、タイプの打鍵誤り、言語情報の伝送途中で発生する誤り、あるいは、音声の機械認識の過程で発生する音韻認識の誤り等の訂正を目的としてすすめられてきた。これまでに提案されてきた方法は、大きく次の2種類に分類できる。すなわち、文字の2字組、3字組、混乱確率、出現頻度等の統計的情報を用いる方法、系列間に距離や類似度を定義して、最もよく似た辞書中の単語を選ぶ方法である。また、接頭辞、接尾辞等の情報を利用するものもある。最近、この種の研究に関するよい解説・展望が出たので^{1)~3)}、ここでは詳説しない。

ところで、綴りの誤り訂正や、それと類似の問題である単語音声の認識では、処理時間をいかにして短縮するかが、最近問題になっている。たとえば、板橋・横山⁴⁾は、単語中の幾つかの音素が定まると、その音素をもつ単語集合を相当小さく指定できると報告している。また、管村・古井⁵⁾は、単語音声認識で有効なダイナミック・プログラミング法⁶⁾の計算量の軽減を図るために、擬音韻標準パターンを用いる方法を提案している。綴りの訂正問題に関連して、栗田・相沢⁷⁾は、置換誤りだけが存在する場合の訂正法を提案し、辞書法⁸⁾よりやや低い訂正率で、数十倍速くできると報告している。

本論文は、文献7)で扱われたのと同じ問題を考え、異なった方法を提案する。すなわち、単語の綴り誤りは、文字の置換誤りだけであるとし、その誤り訂正を高速に行おうとするものである。OCRの文字読取り誤りやタイプの打鍵誤りは、一般に、ある文字がすべての文字に誤る可能性はないことを利用して、探索すべき辞書の範囲を有効に縮小し、時間の短縮とともに、訂正率の向上も図る。2章で高速化法1、3章で高速化法2を述べる。4章で実験結果を述べ、全辞書法と比較する。

2. 高速化法1

Σ を文字の集合とする。一般に、文字読取機はその機械に固有の読取り誤りの癖をもっている。この癖は混乱行列(confusion matrix)で表されることが多い。いま、混乱行列が与えられたとしよう。 Σ に属する二つの文字 x, y について、 x が y と誤って読まれる可能性があるとき、 $x \rightarrow y$ と書くことにする。 $x \rightarrow y$ のとき、 x と y は同じ類(その類を仮に $C(x)$ と書く)に属しているとする。 $u \in C(x)$ 、かつ $u \rightarrow v$ (あるいは、 $v \rightarrow u$)なら、 $v \in C(x)$ であるとする。荒川⁹⁾の手書き英小文字読取り実験では、次の9類に分類できる。

$$\begin{aligned} A &= \{a, h, k, m, n, p, u\}, \\ B &= \{b, e, f, g, l, q, y, z\}, \\ C &= \{i, j, t\}, D = \{o, v\}, \\ E &= \{r, s\}, F = \{w\}, G = \{x\}, \\ H &= \{c\}, I = \{d\} \end{aligned} \quad (2.1)$$

ここで、A, B, ..., I等を類名と呼ぶことにする。このとき、ある文字が、その類に属する文字に誤って読

† High Speed Error-Correcting Methods for Substitution Errors in Spellings by EIICHI TANAKA, TAKAHIRO KOHASHIGUCHI and KUNIHICO SHIMAMURA (Department of Information Science, Faculty of Engineering, Utsunomiya University).

†† 宇都宮大学工学部情報工学科

まれる可能性はあるが、類外の文字に誤って読まれることはない。上記の類の場合、'apple' が 'upple' と読まれることはあっても、'opple' と読まれることはない。いま、 $\alpha = \text{apple}$ が誤って、 $\alpha' = \text{upple}$ と読まれたとする。 α を類名で書くと、 $C(\alpha) = \text{AAABB}$ であり、 α' も $C(\alpha') = \text{AAABB}$ である。 $C(\alpha), C(\alpha')$ を、それぞれ、 α, α' の類名表記と呼ぶ。正しい綴りと誤った綴りは、つねに同じ類名表記をもつ。

ところで、辞書 D が与えられたとき、同じ類名表記をもつ単語をひとまとめにして、辞書中の単語を分類することができる。このとき、類名表記 C_i をもつ単語の集合を $D(C_i)$ と書き、 $D(C_i)$ を「見出し C_i をもつ部分辞書」と呼ぶ。このとき、

$$D = \bigcup_{k=1}^n D(C_k)$$

$$D(C_i) \cap D(C_j) = \phi \quad (\text{空集合})$$

$$(i \neq j, 1 \leq i, j \leq n) \quad (2.2)$$

である。ここで、 n は辞書 D の単語の異なる類名表記の数である。

いま、辞書にある単語 β の幾つかの文字が誤って綴り β' になったとする。 β' を訂正することを考えよう。 β' を類名表記 $C(\beta')$ に変換する。 $C(\beta) = C(\beta')$ であるから、 $C(\beta')$ を見出しにもつ部分辞書 $D(C(\beta'))$ を探す。 $D(C(\beta'))$ の中に必ず β があるから、 D のすべての単語を訂正候補に選ばなくても、 $D(C(\beta'))$ の単語だけを考慮の対象にすればよい。辞書を用いて訂正する方法には幾つかあるが¹⁾、そのいずれの場合にも、上記の議論は成り立つ。いま、辞書を用いて綴り誤りを訂正する方法のうち、最も単純なハミング距離 (HD) を用いる方法を選んだとする。辞書 D のすべての単語を候補とする場合 (全辞書法と呼ぼう) と、本方法で候補対象を縮小した場合を比較してみよう。まず、つねに、

$$|D| \geq |D(C(\beta'))| \quad (2.3)$$

である。いま、

$$\bar{D} = D - D(C(\beta')) \quad (2.4)$$

と置く。いま

$$HD(\beta', \beta) = d,$$

$$\text{Min} = \min \{HD(\beta', \delta) \mid \delta \in \bar{D}\}$$

$$\bar{d} = \min \{HD(\beta', \gamma) \mid \gamma \in D(C(\beta')), \gamma \neq \beta\} \quad (2.5)$$

とする。

(i) $\bar{d} > d$ なら、本方法で訂正される。このとき、全辞書法では、次の場合がある。

a) $\text{Min} < d$ のとき

イ) $\text{Min} = HD(\beta', \delta)$ となる δ が \bar{D} に一つるとき全辞書法では誤訂正される。

ロ) $\text{Min} = HD(\beta', \delta)$ となる δ が \bar{D} に、少なくとも二つあるとき、全辞書法では棄却される。

b) $\text{Min} = d$ のとき、全辞書法では棄却される。

c) $\text{Min} > d$ のとき、全辞書法でも訂正される。

(ii) $\bar{d} < d$ かつ $\bar{d} = HD(\beta', \gamma)$ となる γ が、 $D(C(\beta'))$ に一つるとき、本方法では誤訂正される。

このとき、全辞書法では次の場合がある。

a) $\bar{d} = \text{Min} = HD(\beta', \delta)$ となる δ が、 \bar{D} に少なくとも一つあるとき、あるいは、 $\bar{d} > \text{Min} = HD(\beta', \delta)$ となる δ が、 \bar{D} に少なくとも二つあるとき、全辞書法では棄却される。

b) その他のとき、全辞書法では誤訂正される。

(iii) $\bar{d} < d$ かつ $\bar{d} = HD(\beta', \gamma)$ となる γ が $D(C(\beta'))$ に、少なくとも二つあるとき、あるいは、 $d = \bar{d} = HD(\beta', \gamma)$ となる γ が $D(C(\beta'))$ に、少なくとも一つあるとき、本方法では棄却される。このとき、全辞書法では次の場合がある。

a) $\bar{d} > \text{Min} = HD(\beta', \delta)$ となる δ が \bar{D} に一つあるとき、全辞書法では誤訂正される。

b) その他のとき、全辞書法では棄却される。

以上の結果をまとめて、次の補題を得る。

【補題 1】

(i) 本方法で訂正されるもののうち、全辞書法では棄却されるものがある。

(ii) 本方法で訂正されるもののうち、全辞書法では誤訂正されるものがある。

(iii) 本方法で誤訂正されるもののうち、全辞書法では棄却されるものがある。

(iv) 本方法で棄却されるもののうち、全辞書法では誤訂正されるものがある。

さらに、全辞書法で訂正されたものは本方法でも必ず訂正されることに注意しよう。全辞書法で誤訂正されたもの、あるいは棄却されたもののうち、本方法では訂正されるものがある (補題 1 (i), (ii)) から、次の補題を得る。

【補題 2】 本方法の訂正率は、全辞書法の訂正率に等しいか、それより高い。

さらに、見出し $C(\beta')$ を探す時間が無視できるときは、(2.3) より、高速化法 1 は、全辞書法より、一般に、速い。

【補題 3】 見出しを探す時間が無視できるとき、高速化法 1 を用いたときの訂正に要する時間は、全辞書

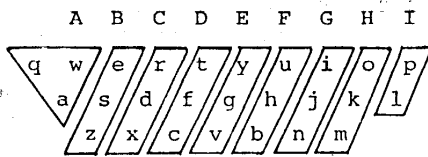


図1 キーボードの鍵の分類
Fig. 1 Classification of keys of a key-board.

法を用いたときのそれに等しいか、それより短い。

3. 高速化法 2

前章の高速化法では、文字誤りの傾向から、文字を複数の類に分類できる、との仮定があった。しかし、そのような仮定がつねに満足されるとは限らない。本章では、前章の仮定をもう少し緩めた条件の下で誤り訂正を考えよう。

タイプの打鍵誤りの場合を例に考える。タイプミスには、(i)原稿の読み誤り、(ii)打鍵誤りが考えられるが、いま、(ii)に限定することにする。キーボードの文字を図1のように、A, B, ..., I の9群に分ける。指の動きから見て、一般に、文字は、それが属する群の他の文字か、両端の群の文字に誤るが、その他の群の文字には誤らないとしてよいであろう。たとえば、A群の文字は、A群の他の文字あるいはB群の文字には誤るが、その他の群の文字には誤らない。B群の文字は、B群の他の文字あるいはA群とC群の文字には誤るが、その他の群の文字には誤らないとする。このとき、前章で述べた方法では、 Σ が一つの類になってしまって、有効な類別ができない。ゆえに、上記の群は、2章の類とは異なった意味のものであるが、用語がいたずらに複雑になるのを避けるために、ここでも類と呼ぶことにする。2章では、誤りは類内でのみ起

表1 混乱行列 (I は入力, O は出力, 表中の空白はゼロ)
Table 1 Confusion matrix (I and O denote inputs and outputs, respectively. A blank in the table denotes 0).

I	O	a	...	d	e	f	...	h	...	l	m	...	o	p	...	t	u	...	z
a																			
b																			
:																			
e																			
f																			
:																			
h																			
:																			
l																			
m																			
:																			

り、本章では、異類間でも誤りが起こるとすれば混乱はない。さて、打鍵誤りと似た事態は、表1のような混乱行列の場合にもある。この混乱行列は、2章の意味では、 Σ が一つの類を作る。いま仮に、

$$G_1 = \{a, b, c, d, e\}, G_2 = \{f, g, h, i, j, k, l\},$$

$$G_3 = \{m, n, o\}, G_4 = \{p, q, r, s, t\},$$

$$G_5 = \{u, v, w, x, y, z\}$$

なる類に分けると、 G_1 類の文字は、 G_1 類の他の文字と G_2 類の文字に誤る可能性があり、 G_2 類の文字は、 G_2 類の他の文字と G_1 類と G_4 類の文字に誤る可能性がある。

いま、一般に、文字を C_1, C_2, \dots, C_m 類に分けたとする。これをもとに、辞書 D の単語を分類し、それから、 $D(C_1), D(C_2), \dots, D(C_m)$ なる部分辞書を作る。辞書にある単語 β の綴りが誤って β' になったとしよう。このとき、 β の類名表記 $C(\beta)$ と β' の類名表記 $C(\beta')$ が一致するとは限らない。この事態に対処するために、次のような重みつきハミング距離 (WHD) を定義する。いま、二つの系列 $X = a_1 a_2 \dots a_m, Y = b_1 b_2 \dots b_m$ の間に、 $\omega(a_i, b_i)$ を

$$\omega(a_i, b_i) = \begin{cases} 1; & a_i \rightarrow b_i \text{ のとき} \\ \infty; & \text{そうでないとき} \end{cases} \quad (3.1)$$

と定義し、次式で定まる値を X と Y の間の WHD とする。

$$WHD(X, Y) = \sum_{i=1}^m \omega(a_i, b_i) \quad (3.2)$$

さて、文字の異類間の誤り数はたかだか u 個であると仮定する。 β' を訂正するために、まず $C(\beta')$ と $\tilde{C} = \{C_1, C_2, \dots, C_m\}$ の元の間で WHD を計算し、 \tilde{C} の元のうち、距離が u 以下のものを選ぶ。これを $C_{s_1}, C_{s_2}, \dots, C_{s_k} (1 \leq k \leq n)$ とする。次に、 β' を $D(C_{s_1}), D(C_{s_2}), \dots, D(C_{s_k})$ の単語との HD を計算し、距離が最小の単語が一つであればその単語に訂正し、そうでなければ棄却する。この訂正法では、同一類内に誤る文字数には制限がなく、異類間の文字誤り数はたかだか u であるとしている。

ところで、一般に、

$$|D| \geq \sum_{i=1}^k |D(C_{s_i})| \quad (3.3)$$

であるから、異類間の文字誤り数がたかだか u であるときは、補題1と補題2は本章の訂正法にもそのまま成立する。

ところで、混乱行列が与えられたとき、どのような文字類に分けるかについては、特に基準はない。しかし、類の分け方によって、訂正手数が大きく影響され

る. 高速化法2では, (i) $C(\beta')$ と \bar{C} の元の WHD を計算する手順と, (ii) β' と $D(C_{s1}), D(C_{s2}), \dots, D(C_{sk})$ の単語との HD を計算する手順がある. いま, 辞書 D の単語数を N , 部分辞書 $D(C_i)$ のサイズを $m_i = |D(C_i)|$ とする. 一つの綴りの訂正に k 個の部分辞書が参照されるものとする. このとき, (i) の手数は, c を定数として, cn (n は部分辞書の数), (ii) の手数は $\sum_{i=1}^k cm_i$ であるから, 全体の手数 L_{ab} は

$$L_{ab} = cn + \sum_{i=1}^k cm_i \quad (3.4)$$

全辞書法の手数は cN であるから

$$N > n + \sum_{i=1}^k m_i \quad (3.5)$$

のときは, 本方法を用いたときに要する計算時間は全辞書法を用いた場合に比べて短い, そうでないときは, 本方法が全辞書法に比べて早いとは言えない. 一般に, 文字類 C_1, C_2, \dots, C_m を多く作れば, 辞書の見出し C_1, C_2, \dots, C_n の数 n は大きくなり, 各部分辞書のサイズは小さくなる. ゆえに, (3.4) の cn が大きくなり, L_{ab} も大きくなる. 逆に, 文字類の数 m を小さくすれば, 見出しの数 n は小さくなり, cn は小さくなるが, 各部分辞書のサイズが大きくなるので, L_{ab} が大きくなる可能性がある.

いま, 各部分辞書のサイズが等しく $m_i = N/n$ ($1 \leq i \leq n$) であると仮定する. このとき, L_{ab} は

$$L_{ab} = cn + kc \frac{N}{n} \quad (3.6)$$

$$\geq 2c\sqrt{kN} \quad (3.7)$$

で, 等号が成立するのは

$$n = \sqrt{kN} \quad (3.8)$$

のときである.

4. 実験結果

提案した二つの高速化法を, 文献10) で使用された6文字単語2,755語を用いて実験した. 使用した計算

表2 見出しとそれに属する単語数の分布
Table 2 Distribution of numbers of words in class name expressions.
A: the number of words in a class name expression. B: the number of class name expressions.

9 類		4 類		2 類	
A	B	A	B	A	B
1	1975	1	734	0~10	10
2	221	2	273	11~20	7
3	49	3	149	21~30	12
4	25	4	74	31~40	7
5	11	5	43	41~50	9
6	6	6	35	51~60	6
		7~8	21	61~80	6
		9~10	11	81~100	2
		11~13	4	101~163	5
合計	2287	合計	1344	合計	64

機は ACOS-600 S である.

高速化法1の実験は, 2章で述べた, 文字誤りに基づく文字の分類が9類の場合 (2.1), $A' = A, B' = B, C' = \{C, D, E\}, D' = \{F, G, H, I\}$ の4類の場合, $A'' = \{A, C, D, E\}, B'' = \{B, F, G, H, I\}$ の2類の場合, の合計三つの場合について行った. 表2に, 各場合の見出し(類名表記)に含まれる単語数の分布を示す. 訂正実験は, 各類について, 単語中の文字誤り数1および2の場合を, 乱数で選んだ250語について行った. 結果を表3に, 訂正例を表4に示す. 文字類の数が多いほど, 訂正率が高くなっている. これは, 予想されるとおりである. 訂正率は全辞書法に比べてかなり高くなっており, また, 訂正に要する時間は $1/30 \sim 1/45$ に短縮されている. 高速化法1の訂正時間は, 主に, $C(\beta')$ を辞書の見出し集合から探す時間と, $C(\beta')$ を見出しとしてもつ単語と β' のハミング距離を計算する時間の和である. 文字類の数が増えると見出しの数が増え, 文字類の数が少なくなると, 一つの見出しの中の単語数が増える. 各類で訂正に要する

表3 全辞書法と高速化法1の比較
Table 3 Comparison between the high speed method 1 and an ordinary dictionary method.

誤りの種類		全 辞 書 法				高 速 化 法 1			
文字類の数	文字誤りの数	訂正率	誤訂正率	棄却率	時間 (msec)	訂正率	誤訂正率	棄却率	時間 (msec)
9	1	0.852	0	0.148	786	0.972	0	0.028	24
	2	0.480	0.028	0.492	786	0.944	0.008	0.048	25
4	1	0.856	0.008	0.136	786	0.956	0.008	0.036	17
	2	0.552	0.032	0.416	786	0.904	0.020	0.076	17
2	1	0.824	0.012	0.164	785	0.884	0.012	0.104	28
	2	0.496	0.052	0.452	786	0.696	0.036	0.268	28

時間が異なるのはこのためである。9類、4類、2類の場合、一つの見出しにある平均単語数は、それぞれ、1.20、2.05、43.05であるから、9類、4類では、主に、見出しを探すのに時間がかかっている。実験では、見出し集合はソートされており、見出し CEFBBC

を見出し集合から探すときには、先頭がCから始まる場所から調べ、時間の短縮を図っている。

高速化法2の実験では、図1の9類を用い、類内の文字、および隣の類の文字に誤るものと仮定した。表5は、見出しに含まれる単語数の分布を示している。実験結果を表6に示す。全辞書法に比べ、訂正率はかなり良くなっている。特に、2文字あるいは3文字誤りの場合の訂正率の向上が著しい。訂正時間は約1/5に短縮されている。訂正例を表7に示す。なお、候補単語数の平均値は、2.00であった。

高速化法1および2の場合とも、探索する辞書の範囲が劇的に小さくなっている。高速化法1が適用でき

表4 誤り訂正例
Table 4 Examples of spelling correction.

正しい綴り β	β を誤らせた綴り β'	β' の類名表記 $C(\beta')$	候補単語 τ	距離 $HD(\beta', \tau)$	判定
recent	receut	EBHBAC	recent	1	訂正
rabble	rablle	EABBBB	rabble	1	訂正
			rubble	2	
			ruffle	3	
			safely	4	
			sagely	4	
sneeze	5				
singer	ringer	ECABBE	ringer	0	誤訂正
			singer	1	
washer	wasner	FAEABE	warner	1	棄却
			washer	1	
parker	parkqs	AAEABE	harper	4	訂正
			marker	3	
			parker	2	
custer	cnrter	HAECBE	carter	1	誤訂正
			custer	2	
fetter	gettes	BBCCBE	better	2	棄却
			fetter	2	
			letter	2	

表5 見出しとそれに属する単語数の分布
Table 5 Distribution of numbers of words in class name expressions.

A: the number of words in a class name expression. B: the number of class name expressions.

9 類	
A	B
1	2535
2	105
3	2
4	1
合計	2643

表6 全辞書法と高速化法2の比較
Table 6 Comparison between the high speed method 2 and an ordinary dictionary method.

誤りの種類		全 辞 書 法				高 速 化 法 2			
類内の文字誤り数	類外の文字誤り数	訂正率	誤訂正率	棄却率	時間 (msec)	訂正率	誤訂正率	棄却率	時間 (msec)
1	0	0.820	0.016	0.164	786	0.920	0.032	0.048	141
2	0	0.532	0.024	0.444	786	0.892	0.016	0.092	142
0	1	0.868	0.012	0.120	785	0.956	0.012	0.032	139
1	1	0.480	0.052	0.468	786	0.912	0.032	0.056	139
2	1	0.112	0.108	0.780	786	0.848	0.060	0.092	138

表7 誤り訂正例
Table 7 Examples of spelling correction (in the case of 2 errors, that is, one is erred to a same class character and the other is erred to another class character).

正しい綴り β	β を誤らせた綴り β'	β' の類名表記 $C(\beta')$	$C(\beta')$ と距離1以内の類名表記	候補単語 τ	距離 $HD(\beta', \tau)$	判定
during	zurijg	BFCGGE	CFCGGE	during	2	訂正
			BGCGGE	ending	4	
				siding	4	
leaden	lesren	IBBCBG	IBACBG	leaden	2	誤訂正
			IBBBBG	lessen	1	
breeze	brewse	FCBABB	ECBABB	grease	2	棄却
			FCBBBB	breeze	2	
			FCBABC	brewer	2	

る場合、文字が2類にしか分類できないような低性能のOCRでも、その出力を訂正する時間が著しく短縮されていることに注目しよう。

5. む す び

(1) OCRの読取り誤りやタイプの手打ち誤りには誤り方に偏りがある。これを綴りの置換誤りの訂正に利用する方法を提案し、英語の6文字語、2,755語について実験した結果を述べた。実験結果を見ると、本方法は、従来の全辞書法に比べて、訂正率が高く、また、訂正に要する時間も、数分の一から数十分の一であった。

(2) 本方法は、辞書を用いて綴り誤りを訂正するとき、探索する範囲を小さくすることによって、訂正率の向上と訂正に要する時間の短縮を図るものである。だから、辞書を用いるすべての方法に応用できる。

(3) 本方法は簡単であり、実用化に有利である。

(4) 長さの短い綴りの訂正にどの程度有効であるかを調べることも、また、長さの変わる誤りの訂正への拡張等が、次の問題である。

謝辞 本研究を行うに当たり、資料をご教示、ご提供いただいたNTT武蔵野研の若原徹氏ならびに、同横須賀通研の荒川弘熙氏、また、日頃ともに勉強している本学鎌田一雄氏に深謝する。

参 考 文 献

1) Peterson, J. L. : Computer Programs for Delet-

- ing and Correcting Spelling Errors, *CACM*, Vol. 23, No. 12, pp. 676-687 (1980).
- 2) 伊藤哲郎: 英文つづり誤りの訂正法, 情報処理, Vol. 25, No. 5, pp. 471-479 (1984).
- 3) 川合 慧: 英文綴り検査法, 情報処理, Vol. 24, No. 4, pp. 507-513 (1983).
- 4) 板橋秀一, 横山晶一: 単語中の音素系列の指定による語彙の減少効果, 信学論, Vol. J67-D, No. 8, pp. 869-876 (1984).
- 5) 管村 昇, 古井貞熙: 擬音韻標準パターンによる大語い単語音声認識, 信学論, Vol. J65-D, No. 8, pp. 1041-1048 (1982).
- 6) Sakoe, S. and Chiba, S. : Dynamic Programming Algorithm Optimization for Spoken Word Recognition, *IEEE Trans. ASSP*, Vol. ASSP-26, No. 1, pp. 43-49 (1978).
- 7) 栗田泰市郎, 相沢輝昭: 日本語に適した単語の誤入力訂正法とその大語い単語音声認識への応用, 情報処理学会論文誌, Vol. 25, No. 5, pp. 831-841 (1984).
- 8) Shinghal, R. and Toussaint, G. T. : A Bottom-up and Top-down Approach to Using Context in Text Recognition, *Int. J. Man-Mach. Stud.*, Vol. 11, No. 2, pp. 201-212 (1979).
- 9) 荒川弘熙: オンライン文字認識に関する研究, 北海道大学学位論文 (1981).
- 10) Riseman, E. M. and Hanson, A. R. : A Contextual Postprocessing System for Error Correction Using Binary n-Grams, *IEEE Trans. Comput.*, Vol. C-23, No. 5, pp. 480-493 (1974).

(昭和60年4月2日受付)

(昭和60年9月19日採録)