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†

栗田 泰市郎†† 相沢 輝昭††

単音節音声認識や OCR を利用して文章や単語を計算機に入力する場合，文字ないし音節の置換誤りを生じる．本論文は日本語の単語入力において，単語辞書と2字組の頻度表などを用いて置換誤りを訂正する方法を提案する．さらにこの訂正法を単音節音声認識と組み合わせて大語い単語音声認識に応用した例について述べる．本文では，まず誤り訂正における日本語と英語の違いについて検討し，日本語の誤り訂正に関する基本的指針を得る．次に，従来の訂正法に誤字の2段階検出などいくつかの改善を加えた訂正法を提案する．基礎的な誤り訂正実験を行ったところ，本訂正法の有効性，とくに2段階検出の効果が確かめられた．模擬単音節認識部と本訂正法による単語認識部(誤り訂正部)を用いての大語い単語音声認識シミュレーションでも良好な単語認識性能が得られた．たとえば，模擬単音節認識部の単音節認識率が95%のとき，単語認識率は約96%，単語認識部での処理時間は約0.3秒であった．本訂正法は他の訂正法の一つである，訂正に際して辞書の全単語を参照する方法に比べ，わずか低い認識率を数十倍速い時間で実現できている．

1. まえがき

単音節音声認識による音声入力や OCR 入力は文章や単語を計算機に入力する有力な手段であるが，入力の際に誤りを生じるため，誤りの訂正などを行ってトータルの認識率を高める研究が古くからなされてきた．また，近年ワードプロセッサの普及に伴い日本語音声入力ワードプロセッサの要望が高まっている．その手段として，かな1文字が1音節にほぼ対応するという日本語の性質を利用して単音節認識がよく用いられているが[12]，ここでも認識精度を高めるため音節の認識誤りを訂正する方法が必要とされ，種々試みられている．

これらの文書あるいは単語データの入力における誤りは置換誤り（ある文字を他の文字にまちがえる誤り）にほぼ限られることが多く，単語内の置換誤りを訂正すれば単語としての認識率を高めることができる．そこで，単語辞書や文字の連接情報などの言語情報を利用した誤り訂正がおもに英単語に関して研究されてきた[1)-8)]．

ここで見られた手法は二つに分類できる[1),4)]．一つは辞書（または単語の見出し表）の探索のくり返し，あるいは辞書に登録されている全単語の参照を基本とする方法である．他の一つは言語において文字の連接のしかたが一様でなく，文字の組合せによって文字組みの出現頻度に大きな差があることを利用する方法である．前者を辞書法，後者を統計法と呼ぶことにする．統計法は言語における文字出現のマルコフ性を利用しているともいえるのでマルコフ法と呼ばれることもある[6),8)]．これらを利用した単語認識システムにおいては，誤り訂正の能力が最終的な単語認識能力に直接大きな影響を与えている．

一般に，辞書法は高い訂正率を得ているが，訂正の対象となる候補がかなり多く，訂正速度は遅い[4),6)]．統計法は，辞書を利用する場合もあるが[3)]，候補の数は少なく訂正速度は速い．しかし，一般に訂正率は低い．

ところで，近年の状況では音声認識や OCR などのデータ入力機器はオンライン的に用いられることが多く，認識の高速化はより重要な課題となっている．単音節認識や OCR を誤り訂正と組み合わせて大語いの単語認識システムを構成できるが，処理の高速性という点では前記の統計法が有利である．しかしながら日本語の単語認識においてこの立場がとられたことは少ないようである[11)]．OCR における阿部らの報告[10)]，音声認識における古井[13)]，外川ら[14)]，中本ら[15)]の報告がみられるが，これらの手法はいずれも辞書法に分類されよう．

本論文ではこのような観点から，統計法の立場にたって日本語に適した誤り訂正法を提案する．さらに，これを単音節音声認識と組み合わせて大語い単語音声認識に応用する．本訂正法はほかにもかな文字を読み取る OCR などと組み合わせることができる．以下

† A Method for Correcting Errors on Japanese Words Input and Its Application to Spoken Word Recognition with Large Vocabulary by TAIICHIRO KURITA and TERUAKI AIZAWA (Information Processing Research Division, NHK Technical Research Laboratories).

†† NHK 総合技術研究所情報処理研究部

2, 3 章では単語の構成要素をかな文字として論じ，4章では音節として論じる．

まず2章では誤り訂正における日本語と英語の違いをおのおのの単語の統計にもとづいて検討し，日本語の誤り訂正に関する基本的指針を得る．

3章では，従来の訂正法に誤字の2段階検出などいくつかの改善を加えた誤り訂正法を提案する．かな鍵盤入力における打鍵誤りを題材に簡単な訂正実験を行い，提案した訂正法の効果を確かめる．

本訂正法は単語辞書，2字組 (digram) の頻度表，入力時の置換誤りに関する confusion matrix を利用して訂正を行うが，これらのデータを記憶する容量は小さい方が好ましいのは当然である．3章ではデータ容量の圧縮についても簡単にふれる．

4章では実際の音声認識装置から得られた認識データをもとに模擬単音節認識部を作成し，それと本訂正法による誤り訂正部（以下では単語認識部と呼ぶ）とにより大語い単語音声認識システムを構成する．このシステムの能力をシミュレーションによって求める．さらにこの結果を辞書法[8]による結果と比較検討する．本訂正法による単語認識部は辞書法によるものと比べ，わずか低い単語認識率を数十倍速い速度で実現できることが明らかになる．

2. 誤り訂正における日本語と英語の違い

統計法による誤り訂正では，データとして n 字頻度表を利用する．n 個の文字を組にしたものを n 字組 (n-gram) と呼び[1]，その種類は文字の種類数を m とすると m^n だけある．n 字頻度表はこれらの n 字組が一般的な文章，あるいは辞書の見出しに現われる頻度を表にまとめたものである．

日本語の 6,460 語からなる辞書* から2文字以下の単語を除いた 5,704 語の辞書を編集し，そのかな見出しに関して n 字頻度表（以下，頻度表と略す）を作成した．このとき，ある文字組が見出しに現われる回数をその文字組の頻度とした．データを**表1**に示す．ここで，濁点，半濁点は1字として扱っており $m=58$ である*．このとき，訂正能力を高めるため，頻度表を単語長ごとあるいは n 字組が現われる文字位置ごとに作成するという方法[7]もあるが，表1においては単語長にも文字位置にも依存しない方法をとった．比較した英語の例は Riseman らによるもので，2,755語の6文字単語が対象となっている[4]．

表1 日本語と英語における n 字頻度表の例
Table 1 Examples of n-gram for Japanese and English words.

言語	日本語		英語[4]	
辞書の登録語数	5,704		2,755	
見出しの総文字数	24,558		16,530	
文字の種類 m	58		26	
n	2	3	2	3
記憶量*	3,364	195,112	676	17,576
ゼロ頻度率	59%	98%	約35%	約85%

* n 字頻度表が記憶すべき頻度の数 ($=m^n$).

表1において，「ゼロ頻度率」は頻度表に記憶される m^n 個の頻度のうち，値がゼロのものの占める割合である．この数値は誤り訂正において重要な意味をもつ．統計法による置換誤りの訂正は，単語内の誤字を探し出す操作（誤字検出）と，誤字を正しい可能性の高い字に置き換える操作（誤字訂正）から成る．これらの過程において，ゼロ頻度率が高いほど誤字を検出できる可能性が高く，誤字訂正において正しい字の候補を絞ることができる．

表1では辞書の登録語数が異なるので単純な比較はできないが，日本語は英語に比べて同じ n ならばゼロ頻度率が高いといえる**．m が大きいためであろう．とくに $n=3$ の場合は 98% という高い値が得られている．しかし，頻度表の記憶量も日本語のほうが大きく，$n=3$ の場合は英語に比べ約11倍となる．$n=2$ の場合，日本語の例はゼロ頻度率，記憶量とも英語の $n=2$ と $n=3$ の中間の値となっている．

これらの結果から，日本語において3字頻度表 (trigram) を利用した訂正は，訂正能力は高いが，データの記憶容量，訂正速度等の訂正にかかるコストもかなり高くなると予想される．Riseman[4]らは頻度表に記憶する頻度を2値化した binary n-gram を用いた．しかしこの結果，約 99% という高い誤字検出力を維持しながら記憶容量が小さくなったが，誤字訂正において数値的な処理ができなくなった．誤りの訂正率は，2,755語の辞書をベースにして最大で 62% と低い値になっている．

われわれは，2字頻度表 (digram) を利用し，誤り訂正のアルゴリズムを強化することによってコストが小さくかつ訂正率の高い訂正法をめざした．もう一つの方向として，$n=3$ のゼロ頻度率 98% に着目し，頻

* ニュース文のかな漢字変換実験用の辞書[10].

* JIS かな符号の文字集合に合わせた.

** ゼロ頻度率は登録語数が多くなるにつれ緩やかに減少する[4]. 暗号学によれば，登録語数よりも見出しの総文字数のほうがより直接的に影響するようである[9].

表 2 単語長の統計
Table 2 Statistics of word length for Japanese and English words.

	例 1[16]	例 2[6]	例 3[7]
言語	日本語*	英語	
語長	語数		
1	139	0	22
2	617	108	89
3	1,288	247	397
4	2,209	467	1,007
5	1,531	571	1,384
6	555	667	1,773
7	96	756	1,886
8	22	624	1,666
9	3	516	1,364
10以上	0	879	2,061
計	6,460	4,835	11,603
平均語長	4.0	7.0	7.3

* 語長はかな文字を単位として数えたもの．このとき濁点，半濁点は1字として扱っている．

度がゼロでない3字組のみを記憶するという方法も考えられる．この場合，3字頻度表へのアクセス時間を増加させないようなデータ構造にすることが重要である．

単音節認識や OCR の後処理として誤り訂正を使用する場合，単語の長さも考慮すべきことである．文字読取り部（OCR）と単語認識部（誤り訂正部）からなる単語認識システムを例にとって説明する．読取り部では各文字を一定の精度で読み取り，その結果を単語ごとに出力する．したがって読取り部分から出力された単語は何文字かの誤字を含んでいる可能性があり，それらを訂正する必要がある．このとき，単語長が短いほど1単語内で発生する誤字の数は少ない．すなわち，訂正すべき文字数が少ない．

表2は日本語と英語の辞書における単語長の統計である．少ない例ではあるが，英語より日本語のほうが平均語長が短いようである．もし読取り部の文字認識率が日本語でも英語でも 90% であれば，平均的な4文字の日本語では1単語内で1字誤る確率は，$0.9^3\times 0.1\times 4\fallingdotseq 0.29$，2字以上誤る確率は $1-0.29-0.9^4\fallingdotseq 0.05$ である．一方，平均的な7文字の英語では1字誤る確率は約 0.37，2字以上誤る確率は約 0.15 である．この例で，日本語では1単語内1字の訂正を行うだけで約 95% までの単語認識率が期待できるのに対し，英語は約 85% しか期待できない．このように日本語の誤り訂正では単語の平均語長が比較的短いため，1単語中1字の訂正能力をもつだけで単語認識システム全体としてかなりの能力が期待できる．

日本語は単音節認識により単語や文章を入力することが可能であるが，単語の平均的な音節数はやはり4程度であり，上と同様な能力が期待できる．このことは後に4章でもふれる．

なお，表1，表2で用いた日本語の辞書は後の実験すべてに使用するので，ここでその性質を調べておく．まず，この辞書のエントロピは3字組 i, j, k の頻度分布を $p(i, j, k)$ などと表すと

単文字に対して

$$F_1=-\sum_i p(i)\log_2 p(i)=4.95$$

2字組に対して

$$F_2=-\sum_{ij} p(i, j)\log_2 p_i(j)=3.59$$

3字組に対して

$$F_3=-\sum_{ijk} p(i, j, k)\log_2 p_{ij}(k)=2.34$$

(bits per letter)

であり，日本語4万字に対する今栄[21]の結果と同様な値を示している．Shannon[22]による英文のエントロピに比べると F_1 は約 0.8 bit 大きく F_2 は同程度である．また F_3 は約 1 bit 小さい．このことは文字を予測するに際し，文字の種類は英文より2倍以上多いにもかかわらず2字組を用いれば予測の候補文字数を英文と同程度に絞ることが可能である，すなわち誤り訂正には有利であることを意味している．3字組の使用はさらに有利であろう．

次に，同じ語長の単語を比較したときそれらの間の異なり文字数 d を単語間距離と定義すると，語数 N の辞書において距離 d にある単語対の数 N_d の N に対する比 $n_d=N_d/N$ は，一つの単語に対しそれと距離 d にある単語の平均個数である[10]．この n_d を前記辞書の3文字以上の単語に関して求めると，

$n_1=18,978/5,704\fallingdotseq 3.3$

$n_2=119,685/5,704\fallingdotseq 35.0$

$n_3=1,336,239/5,704\fallingdotseq 234.3$

となった．これらの数が大きいことは誤り訂正において後に述べる誤訂正が多くなりやすいことにつながり好ましくない．本辞書は語数 N が大きいため阿部ら[10]や中本ら[15]が実験で用いたものよりも n_d が大きく，きびしい条件になっている．

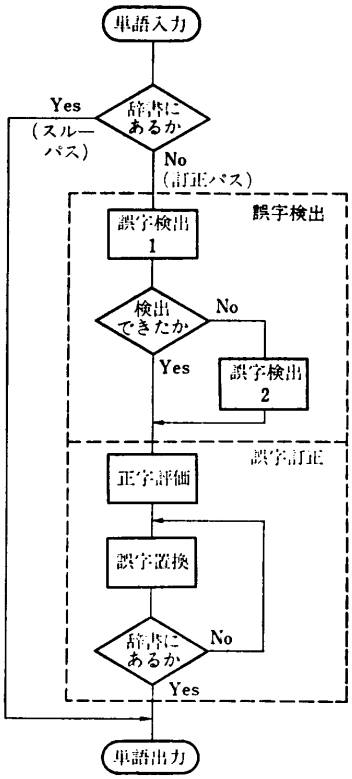

図 1 誤り訂正の流れ
Fig. 1 Flow of the correction of a substitution error per word.

3. 誤り訂正法とその能力[17),18)]

1単語中1文字の置換誤りを訂正する方法について述べる．訂正されるべき単語は単語辞書に登録されているとする．訂正処理の大まかな流れを**図1**に示す．

単語が入力されるとまず辞書を探索する．単語が辞書に登録されていればそれは正しいと判断され，図1のスルーパスを通ってそのまま出力される．登録されていなければ，訂正パスに入り誤り訂正がなされる．このように訂正に先だって辞書を探索する方法はCornew[3)]にも見られる．

3.1 訂正アルゴリズム

訂正パスは誤字検出と誤字訂正の二つの操作から成る．誤り訂正に使用するデータは2字頻度表および音節ないし文字を入力する際の confusion matrix である．confusion matrix はある文字をある文字に認識する m^2 個の頻度を記憶している．

(1) 誤字検出

誤りのある単語内から頻度がゼロである n 字組を見いだすことは誤字を検出する一つの方法である[4)]．n 字頻度表は辞書の見出しに関して作成されているので，ある n 字組の出現頻度がゼロであれば，n 字組は辞書の見出しに現われない，つまり，その n 字組のなかに誤字があることがわかる．これは一つの確実な誤字検出法である．この方法を以下，ゼロによる検出と呼ぶ．一つの頻度ゼロで n 個の誤字（の候補）が検出される．

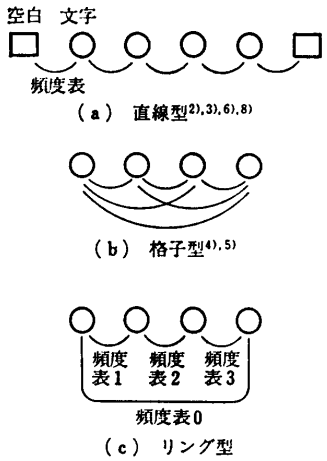

図 2 2字頻度表の適用法 ($L=4$ の例)
Fig. 2 Application methods of digram (example of word length=4).

図1における「誤字検出1」はゼロによる検出を行っている．しかし，3.2節の実験結果で明らかになるように，2字頻度表を利用したゼロによる検出では誤字を検出できる率が低い．そこで，誤字検出1の後に誤字が検出できたかどうかをチェックし，できていなければ「誤字検出2」を行う．この構成を誤字の2段階検出と呼ぶことにする．

誤字検出2では，2字組の頻度値にしきい値 D_{th} を設け，頻度が D_{th} 以下ならばその2字組のなかに誤字があると判断する．$D_{th}=0$ ではゼロによる検出と等価になる．D_{th} を変化させたときの状況は3.2節で論じられる．

ところで，2字頻度表を単語に適用するに当たり**図2**(a),(b)に示すような方法が試みられている．(a)は単語の前後に空白（ブランク）を仮定し，頻度表を直線的に適用する方法であり，2字頻度表をたとえば1文字目と2文字目の頻度表というように文字位置ごとに用意するとして，$L+1$ 枚の頻度表が必要である (L は単語長)．(b)は頻度表をすべての文字の組合せに適用する方法で，${}_LC_2$ 枚の頻度表が必要となる．図ではこれらを仮に直線型，格子型と呼ぶ．

(b)は強力であるが頻度表の記憶量が大きい．たとえば，われわれの辞書には9文字の単語まで含まれて

いるので ${}_9C_2=36$ 枚の頻度表が必要となる．(a)は誤字の検出能力が低い．とくに，単語の語尾と語頭に誤字があるときに検出能力が低いようである．語尾と語頭の2字組の片側はつねに空白であり，情報量が不足するのではないだろうか．

ここでは図2(c)に示す適用法を用いた．単語の語尾と語頭に関する頻度表（図の頻度表0）を用意し，リング型に適用する．必要な頻度表の枚数は L 枚である．これにより，語頭や語尾も他の文字位置とまったく同様な処理が可能になる．これは次の誤字訂正においても有利である．確実な検出である誤字検出1で検出できる率は語尾において8％程度増加する．

(2) 誤字訂正

まず，ある文字が誤字に対してもとの正しい字（以下，正字と呼ぶ）である可能性を評価する．図1の「正字評価」である．誤字検出で検出された誤字が e 個あるとし，文字の種類を m とすると，1単語中に真の誤字は1字しかないと仮定しているので，評価すべき組合せは，$e\times(m-1)$ 通りとなる．

評価値 g は次式で表されるものを用いた．頻度表の適用法はさきのリング型を用いている．

$$g_{ij}=K_{ij}^2\times f_{1ij}\times f_{2ij} \qquad (1)$$

ここで，文字の集合を $\{C_1, C_2, \cdots, C_m\}$，
入力された単語を $W=X_1X_2\cdots X_L$，
検出された誤字の位置を $p_i(1\leq i\leq e)$ とすると，
検出された誤字は X_{p_i} で表せる．

X_{p_i} に対する正字の候補：$C_j(1\leq j\leq m$，ただし，$X_{p_i}\neq C_j)$

g_{ij}：C_j が X_{p_i} の正字である可能性の評価

K_{ij}：音節ないし文字を入力する際に C_j を X_{p_i} に誤る頻度

f_{1ij}：2字組 $\begin{cases}X_{p_i-1}C_j\ (p_i\neq 1)\text{ の頻度}\\ X_LC_j\ (p_i=1)\end{cases}$

f_{2ij}：2字組 $\begin{cases}C_jX_{p_i+1}\ (p_i\neq L)\text{ の頻度}\\ C_jX_1\ (p_i=L)\end{cases}$

g_{ij} をすべての組合せについて求め，g_{ij} の大きい順を正字の候補（以下，正字候補と呼ぶ）の順位とする．K_{ij}, f_{1ij}, f_{2ij} のいずれかがゼロであることにより，多くの文字が候補から除外される．f_{1ij}, f_{2ij} に関してリング型の適用法を用いているため，語尾や語頭の文字でも他の文字位置と同程度に正字候補の数を絞ることができる．

図1の「誤字置換」では，第1の正字候補から順に誤字と置換し，そのたびに辞書を探索する．置換の結果できた単語が辞書に登録されていればその単語を訂正結果として出力する．これを登録されている単語が生成できるまでくり返して誤りを訂正する．

(1)式において，K_{ij} は confusion matrix から得ているが，K_{ij} の2乗としたのは，(1)式における2字頻度表と confusion matrix の相対的な影響力を同等にするためである．われわれのある実験では K_{ij} を用いたとき誤りの訂正率86.7%に対し K_{ij}^2 を用いたとき訂正率88.7%となって K_{ij}^2 のほうが若干よかった．K_{ij}^3 以上では K_{ij}^2 と大差ない結果を得ている．

正字候補の順位について，Cornew[3]は誤字の位置に関してまず順位をつけ，その範囲内で正字候補の評価値によって順位をつけた．つまり誤字の位置と評価値で2段階に順位をつけている．これに対し本訂正法は評価値のみで順位をつける．両者を実際に比較したところ，本方法のほうが訂正率は数%高く，訂正速度は4割程度速かった．

以上の訂正アルゴリズムの形式的記述を付録1に示す．

3.2 訂正実験

JIS 配列のかな鍵盤から単語を入力する際の誤り，つまりキーの押しまちがいを対象に誤り訂正実験を行った．よくいわれるように，鍵盤入力においては置換誤りは誤りの一部にしかすぎない．しかし，本訂正法の訂正能力を知る基礎的な実験であるので，身近な鍵盤入力の置換誤りを対象とした．

単語辞書は表1のときに用いた5,704語のものを使用した．2字頻度表は辞書のかな見出しから図2(c)の方法に従って9枚作成した*．このとき9枚の頻度表のゼロ頻度率はそれぞれ70%以上となっている．

confusion matrix は次のように作成した．通常，打鍵時には隣接キーにまちがえやすい．そこで単純に，あるキーの回りのキーに誤る頻度を 1，他は0とした．

実験は，単語を辞書から選び出し，confusion matrix の作成と同じ基準で選んだ誤字1字を単語内の1字と置き換えて訂正プログラムに入力する．このような試行を5,704語の中から乱数によって選んだ単語100語に対し**，可能な限り誤字と誤字位置を変えて

---

* 辞書に登録されている最も長い単語が9字である．

** このときの単語100語の集合におけるエントロピは一例として
$F_1=4.9$, $F_2=2.4$, $F_3=3.1$ (bits per letter)
であった．また単語間距離に関する統計は一例として
$n_1=0.1$, $n_2=0.6$, $n_3=4.0$
であった．

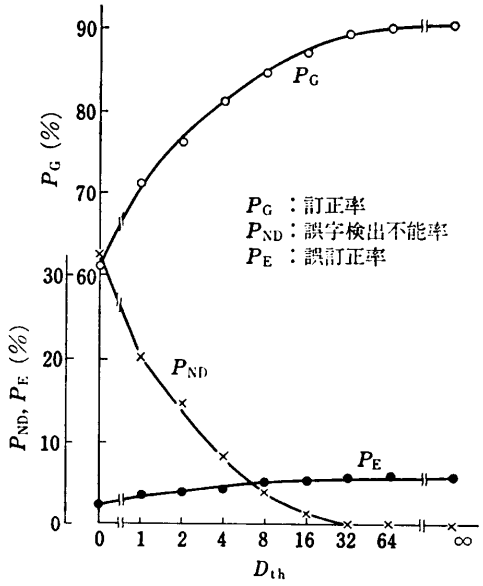

図 3 2段階検出による訂正率の改善
Fig. 3 Improvement of correction rate by the 2-step error detection (increasing the threshold D_{th} of digram).

表 3 誤り訂正の例
Table 3 Examples of spelling correction for Japanese words.

状態	もとの単語	誤った単語	訂正結果
訂正可能	カイトウ	カテトウ	カイトウ
誤字検出不能	クミチガイ	クミチガテ	—
誤訂正	カイトウ	カイテウ	カイテイ
無訂正	テンラン	テンラク	テンラク

合計約 2,000 回行った.

結果を要約すると，単語認識部の出力は**表3**に例を示したような四つの状態に分類でき，その割合は D_{th} の増加に従って**図3**のように変化する.

表3において，誤字検出不能は誤字をまったく検出できなかったため単語がリジェクトされたことを示す．誤訂正はもとの単語とは異なる別の「正しい」(辞書に登録されている) 単語を出力したことを示す．無訂正は誤字を含んでいる状態がすでに「正しい」単語となっているので図1のスルーパスを通って入力がそのまま出力されたことを示す．訂正プログラムの訂正率は訂正可能が全体に占める割合である.

図3において，$D_{th}=0$ の状態，つまりゼロによる検出のみで誤字検出を行った場合，訂正率は 61.3% とかなり低い値になる．訂正可能以外の大部分は 32.5% の誤字検出不能であり，D_{th} の増加により訂正率の改善が期待できる.

表 4 $D_{th}=\infty$ のときの訂正結果
Table 4 Correction rate and other results for $D_{th}=\infty$.

訂正可能	90.2%
誤字検出不能	0
誤訂正	6.1%
無訂正	3.7%
辞書探索数	2.6

誤訂正は最も好ましくない状態といえる．表3の例のように訂正結果から誤りがわかりにくくなることが多いためである．D_{th} を増加させる際に心配されたことは，誤字検出不能が減るかわりに誤訂正が増すことであった．しかし，図3に見るように誤字検出不能の急速な減少は大部分が訂正可能にかわり誤訂正の増加は少なかった.

$D_{th}=\infty$ のとき訂正率は最高となった．**表4**はこのときの各状態の割合である．訂正率は 90.2% と高い値になっている．$D_{th}=\infty$ ということは，ゼロによる検出ができなかった場合は単語内のすべての文字に誤字の可能性があると判断する，ということである.

表4の辞書探索数は辞書を探索する回数の1試行当りの平均であり，訂正速度の一つの目安である．図1の最初の探索があるため，この値は必ず1以上になる．辞書探索数が小さいことは訂正速度がはやいことにつながり，それはまた(1)式の g_{ij} の有効性の一つの目安でもある．本実験といろいろ条件は異なるが，参考として，Cornew[3]は約 70% の訂正率を約 10 回の辞書探索で得ている．本実験では 2.6 回という少ない辞書探索で約 90% の訂正が可能となっている.

なお，表4の結果は辞書から単語を選ぶ乱数を変えてもほとんど変わらない．たとえば，ある別の乱数のとき訂正率 90.6%，誤訂正 5.9%，無訂正 3.6% であった.

3.3 2字頻度表の圧縮

前節では文字位置ごとに9枚の2字頻度表を使用しており，各頻度表のゼロ頻度率は 70% 以上であった．図2(c)にならってこれらを頻度表0～8と呼ぶ．一つの頻度は2バイトで記憶しており，頻度表全部の記憶容量は $58^2\times9\times2\fallingdotseq59.1$ kバイトである.

この量はやや大きいので頻度表 0，および頻度表 1～8 を1枚に圧縮した頻度表 1′ の2枚を使用することにし（つまり，頻度表 0 と 1′ の 2 枚で図2(c)のリングを構成する），また一つの頻度を1バイトに圧

縮して*，訂正実験を行った[18]．この状態で頻度表の記憶容量は $58^2\times2\times1\fallingdotseq6.6$ k バイトとかなり小さくなっている．このように圧縮しても訂正能力はほとんど変わらないことが報告されている[18]．

ここで頻度表 1′ は表 1 に日本語の $n=2$ の例として示したものと同じものである．この頻度表のゼロ頻度率は表 1 から 59% であるが，このようにゼロ頻度率が低下しても訂正アルゴリズムの改善により訂正率の悪化は小さくなっている．

頻度表以外のデータについては，confusion matrix の記憶容量は 3.3 k バイトであり頻度表と同等以下である．辞書はその大きさと用途の広さから大容量の 2 次記憶に置くものと考えている．

文献 18) では，上の圧縮と同時に誤字の発生条件を 3.2 節より複雑な条件に変えているが，これによっても訂正能力はあまり劣化していない．

4. 大語い単語音声認識への応用

4.1 模擬単音節認識部

この章では前章で述べた誤り訂正法を大語い単語音声認識に応用することを試みる．以下では単音節認識部と，音節認識の誤りを単語ごとに訂正して単語を認識する単語認識部からなる大語い単語認識システムを考える．単音節認識部は音声信号を受けて音節の第 1 候補の列を単語ごとに出力し，単語認識部はその列の中に認識誤りがあれば訂正して出力する．図 1 の誤り訂正プログラムを単語認識部としてそのまま用いるので，訂正できるのは 1 単語中 1 音のみである．

単音節認識部として実際の装置を用いるのでなく，以下に述べるような模擬単音節認識部を用いる．

専用の単音節認識装置が入手できなかったので，市販の単語音声認識装置に音節を一つの単語として認識させてみた．この装置は特定話者用のものである．五十音と濁音，半濁音を含む 68 音節を男性 10 人，女性 10 人に 5 回ずつ発声してもらい，装置の認識特性を調べた．この装置の単音節認識率は 62.2% であった．

このとき得られた confusion matrix の対角成分(音節が正しく認識できることを示す成分)に定数をかけると，音節の認識誤りの性質をそのままにして単音節認識率を任意に設定できる．これを模擬単音節認識部とした．前記装置の子音に関する confusion matrix を付録 2 に示す．

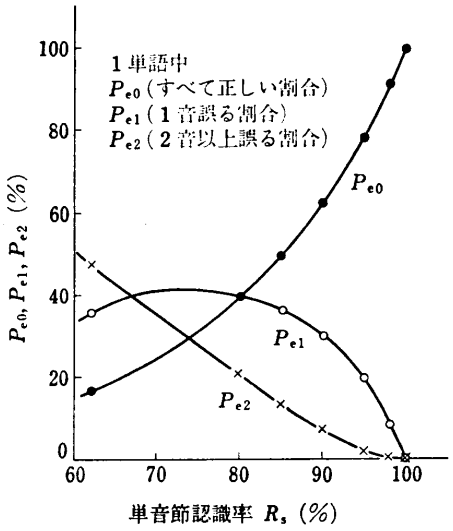

図 4 模擬単音節認識部の特性
Fig. 4 Performance of the simulated syllable recognition block vs. Japanese syllable recognition rate R_s.

図 4 は模擬単音節認識部のみで単語認識シミュレーションを行った結果である．以下では単音節認識率を R_s で表す．R_s は 68 音節を均等に発声した場合に音節認識結果の第一候補が正解である確率である．図において，実用的な $R_s=90\%$ 以上の領域では，1 単語中 1 音誤る割合が誤りの大部分を占めている．$R_s=90\%$ のとき 1 音誤り (図の P_{e1}) は約 30%，誤りのない場合 (P_{e0}) は約 63% であり，1 音誤りを訂正することにより 93% までの単語認識率が期待できる．これは訂正する前の 63% に比べかなり大きな値である．$R_s=95\%$ では単語認識率 98% まで期待できる．

4.2 単語認識シミュレーション

上記の模擬単音節認識部と誤り訂正とにより大語い単語音声認識シミュレーションを行った．

3 章と同様に 6,460 語の辞書から 2 音節以下の単語を除いた 5,496 語の辞書を編集した．そして，この辞書の見出しの音節表現に関して 2 字頻度表を作成した．つまり，68 音節あるので 2 音組の種類は 68^2 である．また，頻度表は 3.3 節で述べたような圧縮された形で使用し，記憶容量は $68^2\times2\times1\fallingdotseq9.0$ k バイトとなっている．誤り訂正に使用する confusion matrix は模擬単音節認識部のものを用いる．

5,496 語の辞書からランダムに，1,000 語を選び模擬単音節認識部に入力する．模擬単音節認識部では confusion matrix にもとづいて乱数により音節の第

* 頻度表 0 と頻度表 1′ は各要素が 2 バイト整数である 58×58 の行列で表しうるが，値が 255 以上となる要素はきわめて少ない (頻度表 0 では 0，1′ では 6 個) ので，255 以上の要素を 255 に置きかえ，各要素を 1 バイト整数に圧縮した．

表 5 単語認識シミュレーションの結果
Table 5 Result of the simulation of Japanese word recognition.

単音節認識率 R_s	90%	95%
P_{e0} (図4参照)	63%	78%
単語認識率	89%	96%
辞書探索数	1.7	1.5
全試行数	1,000	1,000
1音誤り	299	199
2音以上の誤り	73	19
訂正可能	260	175
訂正率*	87%	88%

* 1音誤りに対する訂正可能の割合

一候補の列を出力する．単語認識部はこれを受け，誤り訂正を行って単語を出力する．誤字の2段階検出における D_{th} は3.2節の結果から $D_{th}=\infty$ とした．

表5は $R_s=90\%$ と $R_s=95\%$ の場合の結果である．表の P_{e0} は図4に示したもので，単語認識部の出力における最終的な単語認識率はその下に示されている．

$R_s=95\%$ では，1単語の認識当り1.5回と少ない辞書探索数にもかかわらず，単語認識率は96%と高い値になっている．1音の置換誤りを訂正できた状態である「訂正可能」は1,000語中175語であり，1音誤り199語に対する訂正率は88%であった．

4.3 辞書法との比較

提案した統計法による単語認識部の性能を辞書法によるものと比較する．

Shinghal は高い認識率が得られる DA (Dictionary Algorithm) 法のリストを示した[8]．これは特別なアルゴリズムによる文字認識部と組み合わせて用いられているが，第一候補のみを出力する単音節認識部に適合させることは簡単である．この DA 法を図1の訂正パスに置いたプログラムを作成した．付録3にその形式的記述を示す．DA 法は誤り訂正に際し，辞書の全単語を参照している．

4.2 節と同様な方法でシミュレーションを行った．試行数は500単語とし，1単語当りの認識時間を CPU 時間により測定した．ここでいう認識時間は単語認識部における処理時間である．計算機は TI の DS 990 ミニコンである．提案した方法および DA 法による結果を**表6**に示す．表6における「認識率」は表5と同様に訂正後の最終的な単語認識率である．

表6から明らかなように，提案した方法は DA 法よりやや低い認識率を30～60倍速い認識速度で実現している．とくに $R_s=95\%$ のときは両者の認識率の差は1%程度にすぎない．このとき提案した方法は1単語を約0.3秒で処理できる．$R_s=90\%$ のときは約4%の差である．また，単語を正しく認識できない場合のなかで最も好ましくない誤訂正も，提案した方法は DA 法と比べ同数以下である．この様子を**表7**に示す．

表 6 提案した方法と DA 法[9] の比較
Table 6 Comparison of the proposed method vs. DA method.[9]

R_s (%)	提案した方法		DA 法	
	認識率 (%)	認識時間 (sec)	認識率 (%)	認識時間 (sec)
80	75	0.48	83	31
85	81	0.39	86	26
90	88	0.36	92	19
95	97	0.29	98	10

表 7 $R_s=90\%$ のときの結果の分析
Table 7 Analysis of the result of Table 6 for $R_s=90\%$.

状態	提案した方法		DA 法	
	1音誤り	2音以上の誤り	1音誤り	2音以上の誤り
訂正可能	128	0	125	22
誤訂正	15	8	18	15
無訂正	8	0	8	0
2音誤りを検出	0	5	0	0
訂正不能	0	24	0	0
訂正率(%)	85	0	83	59

表7は1音以上誤りがある場合に対する認識結果の分析である．表に示したのは $R_s=90\%$ の場合であるが，他の場合でも同様な傾向を示す．この場合，500単語中で模擬単音節認識部が1単語につき1～3音誤る割合はそれぞれ，151，32，5語だった．4音以上誤ることはなかった．表の「2音誤りを検出」と「訂正不能」は，それぞれの理由により単語がリジェクトされた状態である．

表7をみると，提案した方法と DA 法の認識率の差は2音以上誤りがある場合（大部分は2音誤りの場合）の訂正能力のみである．したがって，1音誤りが誤りの大部分を占めているとはいえ，図1の単語認識部に2音誤りの訂正機能を加えれば，認識能力はさらに改善できよう．1音誤りに対する訂正率はわずかではあるが提案した方法のほうがまさっている．

DA法が2音誤りに対して訂正能力をもっているとはいえ，その訂正率は約60%と低い値である．この程度の訂正率は図1の誤り訂正アルゴリズムをそのまま2音誤りに拡張しても達成できるのではないかと考えられる．具体的には，誤字訂正において正字候補の評価値を誤字を1単語内に2字仮定してさまざまな組合せについて求め，評価する方法などが考えられる．この点は今後の検討課題である．

なお，表6の認識時間は R_s が小さくなると増加するが，これは R_s が小さくなると図1の訂正パスに入る割合が増えるためである．訂正パスに入った場合の処理時間の平均は R_s によらずほぼ一定で，提案した方法で約1秒，DA法で約50秒であった．

ところで，最近，文字の連接情報を認識された音節や文字の下位候補にまで適用し，日本語の単語認識率を向上させたという報告がなされている[19),20)]．本方法においても，下位候補や音節間の距離データを利用すれば図1の誤字訂正において正字候補をより絞れることが期待でき，表7の誤訂正はさらに減るであろう．下位候補や距離データの利用も提案した方法の訂正能力ひいては単語認識能力を改善できると考えられる．

5. むすび

日本語を入力する際に生じる置換誤りの訂正法とその大語い単語音声認識への応用について述べた．

本訂正法は文字の連接情報を利用して訂正を行う統計法に分類されるものである．従来の方法にいくつかの改善を加えた結果，基礎的な訂正実験で高い訂正率が得られた．とくに，誤字検出における2段階検出の採用は，日本語の性質とあいまって，高い訂正率を記憶容量の小さい2字頻度表 (digram) を用いて実現することを可能にした．

模擬単音節認識部を用いての大語い単語音声認識シミュレーションにおいては，単音節認識率が95%のときに，96%程度の単語認識率が認識時間約0.3秒で得られた．これは，辞書法による認識シミュレーションの結果に比べ，認識率は約1%低く認識速度は約30倍速い結果となっている．両者に認識率の差が出る原因を分析したところ，それは1単語内に誤りが2音ある場合の訂正能力であることがわかった．1単語内に誤りが1音のときの訂正率はほぼ等しい．

ここで報告した単語認識部（誤り訂正部）は単音節認識部からの情報として単音節の第一候補のみしか用いていない．下位の候補とそれらの音節間距離データを利用すれば，より高い単語認識能力が期待できる．現在，これらの情報を利用して1単語内に誤りが2音ある場合まで訂正できる訂正法を検討している．

単語辞書を利用した単語認識システムの性能は辞書の登録語数が大きくなると低下することが知られている[4),7),13),14)]．認識シミュレーションで用いた辞書は5,496語から成る．この大きさは過去の報告で用いられた辞書の大きさに比べて小さくはないといえよう．数万語から成る汎用辞書を用いて認識能力を測る必要がある一方で，その大きな辞書を分野別に分類するとすれば各分野別の辞書の大きさは数千語になるだろうという考え方もある．辞書を分野別に分類することは，かな漢字変換など他の言語処理においても有用である．現在，NHK編新用字用語辞典をもとに分野別辞書の作成を進めている．

謝辞 おわりに，日ごろご指導いただくNHK総合技術研究所情報処理研究部の町田部長，沓沢主任研究員に深く感謝する．

参考文献

1) 川合 慧：英文綴り検査法，情報処理，Vol. 24, No. 4, pp. 507-513 (1983).
2) Harmon, L. D.: Method and Apparatus for Correcting Errors in Mutilated Text, U. S. Patent, No. 3188609 (June 8, 1965).
3) Cornew, R. D.: A Statistical Method of Spelling Correction, *Inf. Control*, Vol. 12, pp. 79-93 (1968).
4) 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).
5) Hanson, A. R., Riseman, E. M. and Fisher, E.: Context in Word Recognition, *Pattern Recogn.*, Vol. 8, No. 1, pp. 35-45 (1976).
6) 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).
7) Hull, J. J. and Srihari, S. N.: Experiments in Text Recognition with Binary *n*-Gram and Viterbi Algorithms, *IEEE Trans. PAMI*, Vol. PAMI-4, No. 5, pp. 520-530 (1982).
8) Shinghal, R.: A Hybrid Algorithm for Contextual Text Recognition, *Pattern Recogn.*, Vol. 16, No. 2, pp. 261-267 (1983).
9) 加藤正隆：基礎暗号学，数理科学，No. 179, pp. 76-83 (1978).

10) 阿部圭一他：辞書を利用する文字認識系の能力の評価，信学論，Vol. 52-C, No. 6, pp. 305-312 (1969).
11) 上坂吉則，黒木総一郎：文字の結合確率の言語認識への利用について，NHK技研月報，Vol. 17, No. 6, pp. 424-434 (1965).
12) 中津良平：音声認識技術，情報処理，Vol. 24, No. 8, pp. 984-992 (1983).
13) 古井貞熙：単音節認識とその大語い単語音声認識への適用，信学論，Vol. J 65-A, No. 2, pp. 175-182 (1982).
14) 外川文雄他：模擬単音節認識による辞書照合の検討，音響学会講演論文集，1-4-2 (1982).
15) 中本豊秋，中川聖一：単音節単位の入力による不特定話者の大語彙単語音声認識，音響学会音声研資，S 82-20 (1982).
16) 相沢輝昭，江原暉将：カナ・漢字変換，NHK技研月報，Vol. 16, No. 7, pp. 260-265 (1973).
17) 栗田泰市郎，相沢輝昭：かな単語入力における誤入力訂正，情報処理学会「自然言語処理技術」シンポジウム論文集，pp. 125-130 (1983).
18) 栗田泰市郎，相沢輝昭：単語辞書を利用した誤入力訂正，NHK技研月報，Vol. 26, No. 10, pp. 354-359 (1983).
19) 杉村利明，斎藤珠喜：文字連接処理によるかなリジェクトの救済，情処全国大会第25回，7B-3 (1982).
20) 大山 裕他：単音節音声からのカナ文字列形成，情処全国大会第26回，1H-6 (1983).
21) 今栄国晴：日本語の digram の相対頻度とその特性，心理学評論，Vol. 4, pp. 85-100 (1960).
22) Shannon, C. E.: Prediction and Entropy of Printed English, *BSTJ*, pp. 50-64 (1951).

付 録 1

提案する訂正アルゴリズムの形式的記述を以下に示す．以下において入力される L 文字の単語を $W=X_1X_2\cdots X_L$，取り扱う文字の集合をその種類数を m として $\{C_1, C_2, \cdots, C_m\}$ で表す．また実行する文を‘{’と‘}’で囲み，コメントを‘〔’と‘〕’で囲む．

誤り訂正アルゴリズム
{単語 $W=X_1X_2\cdots X_L$ を入力する}；
if (W が辞書にある) then go to Step 3；
Step 1：〔誤字検出〕
{2字組 X_LX_1 が辞書の見出しに現われる頻度 h_0．
2字組 $X_1X_2, X_2X_3, \cdots, X_{L-1}X_L$ の頻度 $h_1, h_2, \cdots, h_{L-1}$ を2字頻度表から求める}；
if ($h_0, h_1, h_2, \cdots, h_{L-1}$ のなかにゼロがある)
then {ゼロの2字組を構成する文字を誤字とする}
else {$h_0, h_1, \cdots, h_{L-1}$ のなかに値が D_{th} 以下のものがあればその2字組を構成する文字をすべて誤字とする}；〔誤字検出2〕
$e \leftarrow$ (誤字の個数)；
if $e=0$ then go to Step 4；〔誤字検出不能〕
Step 2：〔誤字訂正〕
for $i \leftarrow 1$ to e do
for $j \leftarrow 1$ to m do
if $X_{p_i}=C_j$ then $g_{ij} \leftarrow 0$
else $g_{ij} \leftarrow K_{ij}{}^2 \times f_{1ij} \times f_{2ij}$；
〔p_i は検出された i 番目の誤字の位置，C_j は j 番目の種類の文字，K_{ij} は単語入力時に C_j を X_{p_i} に誤る頻度，f_{1ij}, f_{2ij} はそれぞれ2字組 $X_{p_i-1}C_j$，$C_jX_{p_i+1}$ の頻度，ただし $X_{p_i}=1$ の場合 f_{1ij} は X_LC_j の頻度，$X_{p_i}=L$ の場合 f_{2ij} は C_jX_1 の頻

付表 1 単音節認識における装置の confusion matrix
Table A. 1 Confusion matrix of the spoken word recognition device for the 68 syllable recognition.

in \ out	(v)	k	s	t	n	h	m	y	r	w	N	g	z	d	b	p	(%)
(v)	227	31	0	10	7	52	11	3	14	4	1	41	2	8	50	39	45.4
k	2	293	3	17	0	61	0	1	1	0	0	77	6	0	9	30	58.6
s	0	1	345	63	0	2	1	0	2	0	0	0	83	1	1	1	69.0
t	3	8	45	255	0	20	3	3	9	1	0	4	83	34	7	25	51.0
n	3	2	0	5	356	7	78	7	21	0	0	2	3	2	5	9	71.2
h	25	46	1	11	2	307	5	1	4	0	0	25	7	2	25	39	61.4
m	6	0	0	2	61	11	360	2	3	2	0	6	0	3	22	22	72.0
y	0	0	0	1	0	0	0	294	2	0	1	0	0	2	0	0	98.0
r	4	5	2	11	15	4	12	15	304	0	0	17	22	27	46	16	60.8
w	0	0	0	0	0	0	4	0	0	92	0	1	0	0	3	0	92.0
N	3	2	0	0	1	3	2	1	2	0	78	1	1	0	3	3	78.0
g	8	72	0	3	0	19	2	6	16	1	0	301	7	25	30	10	60.2
z	2	1	56	48	0	1	0	1	6	0	0	2	368	7	5	3	73.6
d	3	4	3	24	7	2	3	6	28	0	0	10	31	161	16	2	53.7
b	12	12	0	6	4	13	14	2	38	5	0	23	14	23	257	77	51.4
p	18	16	3	21	3	25	8	0	17	1	0	10	6	14	98	230	52.0
																	62.6

```
    度を表す〕
    {g_ij を大きい順にソートする};
    {g_ij がいちばん大きい C_j と X_pi の組について X_pi
      を C_j に置換する};
    while (W が辞書にない) do begin
      {前に置換した C_j を X_pi にもどす};
    {g_ij が次に大きい C_j と X_pi の組について X_pi を C_j
      に置換する}
    end;
Step 3: {単語 W を出力する};
Step 4: end. 〔訂正アルゴリズム終了〕
```

付 録 2

4.1 節で述べた市販の単語音声認識装置を単音節認識に用いた場合の子音に関する confusion matrix を**付表1**に示す．子音認識率は 62.6% であり，単音節認識率 62.2% に近い．すなわち音節認識誤りの原因はほとんど子音の誤りである．母音認識率は 98.5% だった．

付 録 3

4.3 節における DA 法[8] の適用法を以下に示す．記法等は付録1と同様である．

```
Dictionary Algorithm
  {単語 W=X1X2…XL を入力する};
  if (W が辞書にある) then go to Step 1;
  for i=1 to N do 〔N は辞書の単語数〕
    if (Wi' の語長)=L then begin
      〔Wi'=Xi1'Xi2'…XiL' は辞書の i 番目の単語〕
      gi←1;
      for j←1 to L do
      gi←Kij×fij×gi
      〔Kij は Xij' を Xj に認識する頻度，fij は2字
      組 Xij'Xi j+1' の頻度，ただし j=L のときは
      Xij'Xi1' の頻度〕
    end;
    W←(最も gi の大きい Wi');
Step 1: {単語 W を出力する};
end.    〔DA 法終了〕
```

(昭和58年11月11日受付)
(昭和59年 4 月17日採録)