

DLM (Digraph Language Model) による文字認識後処理

中村 匡伸 紺野 章子 本郷 保夫
富士ファコム制御(株)

あらかし 本稿では、言語シンボルを有向グラフにおける「点」、シンボル同志の接続を「枝」、単語を「道」と捉えた有向グラフ言語モデル(DLM: Digraph Language Model)を提案し、それを用いた文字認識後処理手法について述べている。DLMの特徴として 1) 入力各文字に対する並列処理が可能、2) 先頭の文字が不明でも単語検索が可能、ということが挙げられるが、今回はまずDLMの辞書である「前後接リスト」(PPAL: Pre-Post Adjacent List)と汎用CPU処理用の単語照合法である Γ 照合法を紹介する。 Γ 照合法は、文字認識結果である候補文字群に対して確からしい文字から単語照合を行う手法であり、それにより不要な候補単語の抽出が抑制され、照合速度の高速化と誤修正の防止を図ることができる。DLM- Γ 照合法を新聞読取結果に適用して漢字を含んだ自立語の照合を行い、文字認識率は95.2%から97.7%になり、処理速度は語彙数10万、4MIPSで2.3 msec/文字となった。

Post-Processing in Character Recognition Using DLM(Digraph Language Model)

Masanobu NAKAMURA Akiko KONNO Yasuo HONGO
Technical Engineering Group FUJIFACOM Corp.
1, Fuji-machi, Hino-shi, Tokyo

Abstract In this paper the new language model, DLM (Digraph Language Model), is proposed, and the post-processing method of character recognition using DLM is described. DLM has the following merits; (1) the parallel processing can be executed to each characters, (2) the word whose head character is even unknown can be searched. Here, PPAL that is the dictionary of DLM and Γ -matching that is the word matching method for a universal CPU, are introduced. Γ -matching has been applied to Japanese newspapers and the recognition rate has become 97.7% from 95.2% and the processing speed was 2.3 msec/character.

1. まえがき

近年、帳票や印刷文書を対象とした文字認識装置が普及してきている。住所、氏名などの単語や文章を文字認識の対象とする場合は、パターン認識結果に言語処理を施すことにより総合的な認識率を向上することができ、各所で研究、開発がおこなわれている。

従来の誤読修正手法は大きく次の2つに分けることができる。

(1) 文字認識の結果リジェクトされた文字に対し、その前後の文字の接続可能性から誤読修正を行う手法^{(2) [6]}

(2) 文頭から逐次的に単語照合と文法検定を行って、誤読修正を行う手法^{(4) [7] [8]}

本稿で提案する有向グラフ言語モデル (D L M : Digraph Language Model) は言語シンボルを有向グラフの「点」、シンボル同志の接続を「枝」、単語を「道」として捉えた言語モデルである。D L M は言語シンボルと単語を有向グラフで体系的に捉えることにより各言語シンボルに対する並列処理が可能になるとともに、先頭文字が不明でも単語の検索を行うことができる。

今回はD L M の辞書である「前後接リスト」(P P A L : Pre-Post Adjacent List) と汎用C P U で誤読修正を行うに適した単語照合法であるΓ照合法を紹介する。Γ照合法は確からしい文字を起点とした単語照合法であり、これにより不要な候補単語の抽出を抑制し、照合時間を短縮するとともに誤修正の防止を図ることができる。

2. D L M (Digraph Language Model) 定義

言語はその素材とも言うべき構成要素である単語とその並べかた、即ち構成規則である文法からなると見ることが出来る。

言語 { 構成要素・・・単語
構成規則・・・文法

文法については句構造文法を始め各種のモデルが提案されている。言語を単語レベルで捉えた従来のモデルとして、アルファベット集合 $S = \{ a_1, a_2, \dots, a_k \}$ の閉包(Closure)がある。これはアルファベットが k 個並んだ集

合 $S^k = \{ a_1 a_2 \dots a_k \}$ の和集合 $S^* = S^0 + S^1 + \dots + S^k + \dots + S^n$ のことであり、各種オートマトンはこの閉包 S^* を処理対象としている。

本稿で提案する言語モデルは言語シンボルを有向グラフの「点」(Vertex)、シンボル同志の接続を有向グラフの「枝」(Arc)として捉える。そうすると単語は有向グラフにおける「道」(Path)で表現される。この言語モデルを有向グラフ言語モデル(D L M : Digraph Language Model)と呼ぶ。D L M は、まず第一に言語素材としての単語の言語モデルを提供する。単語レベルよりさらに高次の、並列性を重視した統語論、意味論を付加することも可能であるが、今回は単語照合のみを行った。D L M を用いると従来では困難だった任意の文字位置からの単語検索を行うことができる。次にD L M の定義を記す。

[D L M 定義]

$D = D(S, A)$:

D は有向グラフ

S は言語シンボル s_i を要素とする点集合

$$S = \{ s_i \mid 1 \leq i \leq u \}$$

A は2つのシンボル s_i, s_j が隣接して語 w を構成し、 s_i が語 w の k 番目の文字であることを示す枝集合

$$A = \{ a_{i,j}^{w,k} \mid 1 \leq i \leq u, 1 \leq j \leq u, 1 \leq w \leq v, 1 \leq k \leq (l_{max} - 1) \}$$

P :

P は語 w をなす点と枝の交互列で表された道の集合

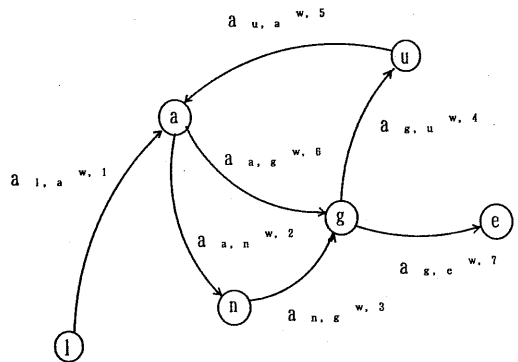


図1. 単語 "language" の有向グラフによる図式表現

$$P = \{ p_w \mid p_w = (s_h, a_{h,i}^{w-1}, s_i, \dots, a_{i,l}^{w-1}, s_l), s_h, s_i, \dots, s_l \in S, a_{h,i}^{w-1}, \dots, a_{i,l}^{w-1} \in A, 1 \leq w \leq v, 1 \leq l \leq l_{max} \}$$

[定義終了]

語長 l の語 w を有向グラフの道 $p_w = (s_h, a_{h,i}^{w-1}, s_i, \dots, a_{i,l}^{w-1}, s_l)$ で表す。 s_h と s_l は道 p_w の始点と終点であり、それぞれ語 p_w に関する語頭シンボル、語尾シンボルと言ひ、両者を合わせて語端シンボルと呼ぶ。 $l=1$ の場合は $p_w = (s_h)$ となり、道が「退化」している状態で1文字単語を表す。言語シンボルの総数は $|S|=u$ であり、語彙総数は $|P|=v$ である。

言語を有向グラフで表現することにより、任意の文字位置から単語を検索することができる。今、ある単語を $s_h s_i \dots s_k \dots s_l$ ($s_h s_i \dots$ は言語シンボル)とすると、従来は単語の先頭文字から順次 $s_h s_i \dots$ と検索していた。DLMによれば単語中の任意の文字 s_k を起点として s_k から出ていく枝(後接シンボルへの枝)をたどることによって語尾方向に、

s_k へ入ってくる枝(前接シンボルからの枝)をたどることによって語頭方向に、単語の検索を行うことができる。

図1に単語"language"の有向グラフによる図式表現を示す。

3. DLM辞書 (PPAL)

有向グラフの一般的なデータ構造として、隣接テーブルと接続テーブルがある。これらは点同志の隣接関係あるいは点と枝の接続関係をマトリックスで示したものであり、この行列値を変更することで単語を表現することが可能だが、語彙数10万の規模ではアドレス空間だけで数十MB~1GBとなりメモリ容量上実用的でない。また、有向グラフの一般的なリスト表現に頂点と枝を表す2つの配列を用い、ある頂点 s_i から出る枝、または s_i へ入る枝を順次ポインタで連結したものがあがるが、これは単語Pathの表現と検索には適していない。

DLMにおいて単語は「道」として捉えられ、「道」の検索・照合に適したデータ構造として「前後接リスト」(PPAL:Pre-Post Adjacent List)を考案した。これは語のPathが

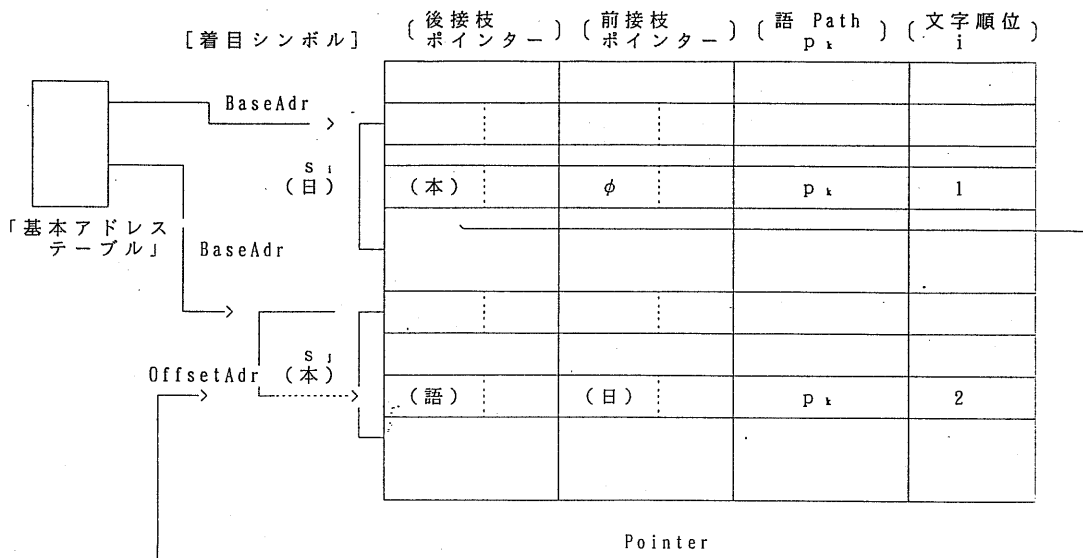


図 2-1 「前後接リスト」

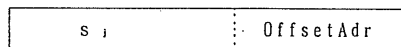


図 2-2 前(後)接枝ポインタ

(入力文)

| | | | | | | | |
|-------|-------|----|-----------|-------|-----------|----|-------|
| s_1 | s_2 | .. | s_{i-1} | s_i | s_{i+1} | .. | s_n |
|-------|-------|----|-----------|-------|-----------|----|-------|

(候補文字)

| | | | | | | | |
|----------|----------|----|-------------|----------|-------------|----|----------|
| c_1^1 | c_2^1 | .. | c_{i-1}^1 | c_i^1 | c_{i+1}^1 | .. | c_n^1 |
| c_1^2 | c_2^2 | .. | c_{i-1}^2 | c_i^2 | c_{i+1}^2 | .. | c_n^2 |
| \vdots | \vdots | | \vdots | \vdots | \vdots | | \vdots |
| c_1^m | c_2^m | .. | c_{i-1}^m | c_i^m | c_{i+1}^m | .. | c_n^m |

図 3-1 入力文 $\{s_i\}$ 、候補文字集合 $C = \{c_i^j\}$

$\{c_i^1\}$

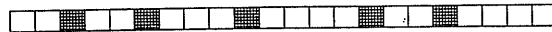


図 3-2 Γ_i 集合

図 3 処理対象の文字集合

次々とシンボルへ入り、出ていく様子を表現したものであり、前後枝または後接枝を順次辿って語頭方向と語尾方向に単語の検索を行うことができる。前後接リストの構造を図 2-1 に示す。(例として、「日本語」の場合をあげる。)

前後接リストは辞書項目が 1 つのシンボル群ごとにまとめてあり、その先頭アドレスである BaseAdr へは基本アドレステーブルにより、そのシンボルをキーとしてアクセスする。辞書項目は一つの単語の中でそのシンボルが、どのシンボルから接続してきたかを示す前接枝ポインタ、そしてどのシンボルへ接続していくかを示す後接枝ポインタ、及びそのシンボルを含む語の Path No. p 、及びそのシンボルが単語内で何番目にあるかの文字順位 i からなる。さらに、後接枝ポインタは後接シンボルそれ自身と単語内の後接シンボルが後接していくシンボル群の先頭アドレスからどの位置に記載されているかを示す Offset アドレスからなる。前接に関しても同様である。尚、語尾シンボルの後接と語頭シンボルの前接には ϕ (NULL) を記載しておく。

今、1 つの単語内のあるシンボル s_i に s_j が後接している場合、 s_i のその単語における辞書項目から次の s_j の辞書項目へは次の様にしてアクセスする。

(α) s_j 群の先頭アドレスを基本アドレス

テーブルで参照する。

(β) 得られた BaseAdr とその語が接続していく辞書項目が s_j 群の先頭からどの位置にあるかをしめす OffsetAdr の和を求め、その単語の s_i に関する辞書項目が記載されているアドレスへの Pointer を作成してアクセスする。

この操作を繰り返すことにより、語尾方向へ検索を行う。語頭方向についても、同様である。辞書の後接シンボルまたは前接シンボルが ϕ の場合、そのシンボルが語尾または語頭シンボルであることを示す。

4. DLM- Γ 照合法

DLM は並列処理に適しているが、OCR の認識率が 90% 以上の読取結果を汎用 CPU で処理する場合は確からしい文字から単語照合を行うとよい。そうすることにより候補文字の組合せから生じる不要な候補単語の抽出を抑制し、照合時間の短縮と誤修正の防止を図ることができる。確からしい文字から単語を照合する手法を DLM- Γ 照合法と呼ぶ。

今、入力文を $S = \{s_i \mid 1 \leq i \leq n, \}$ 、候補文字を $C = \{c_i^j \mid c_i^j \text{ は } s_i \text{ の第 } j \text{ 位の候補文字, } 1 \leq i \leq n, 1 \leq j \leq m\}$ とする。(図 3-1) 始めに、1 位の候補文字 $C^1 = \{c_i^1\}$ を候補文字の類似度や統計的データ

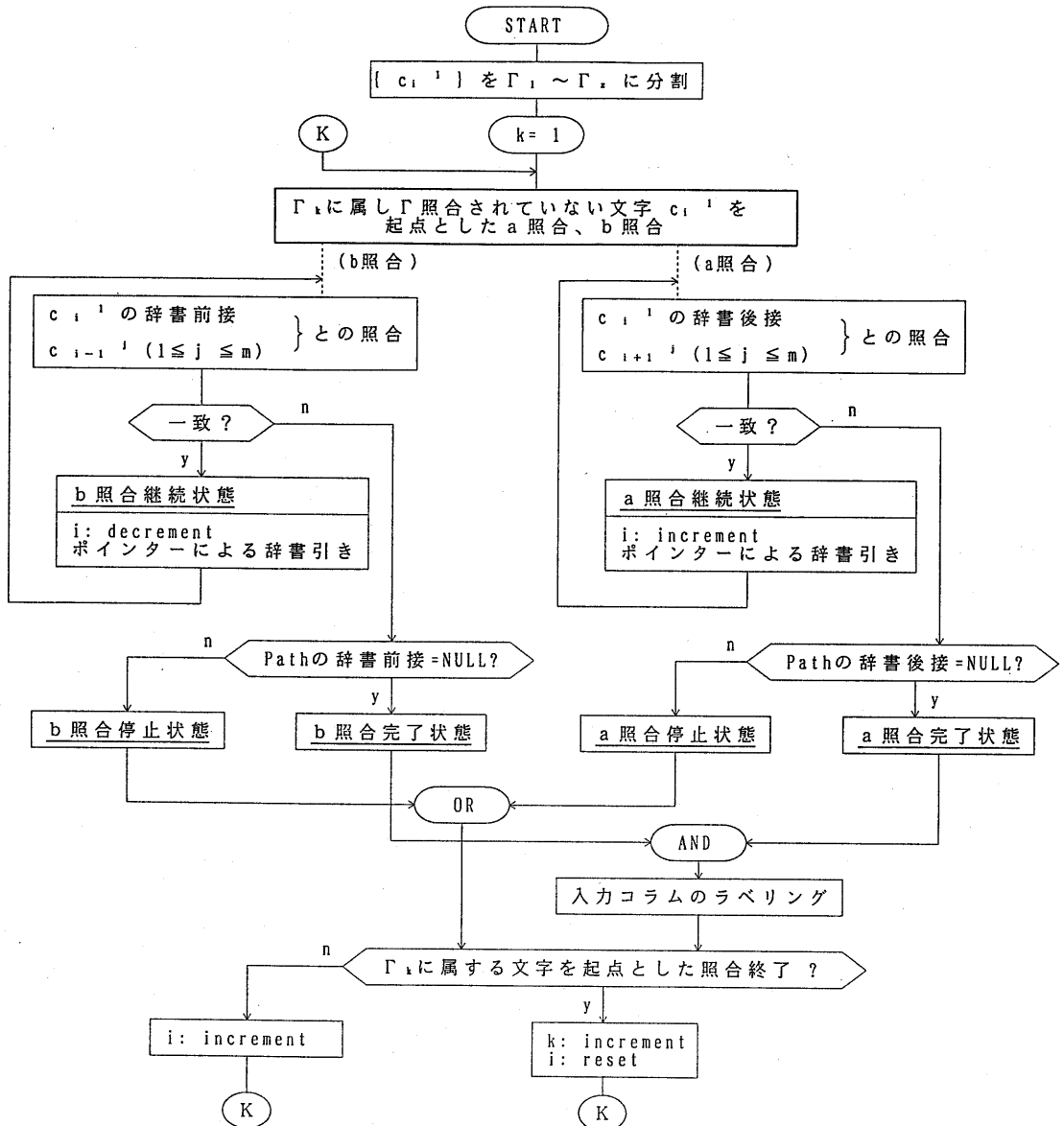


図4 DLM-Γ照合法 処理フロー

等から正解である確率の高い順に $\Gamma_1, \Gamma_2, \dots, \Gamma_m$ へ分割する。 $(\Gamma_1 \cup \Gamma_2 \cup \dots \cup \Gamma_m = C^1, \Gamma_1 \cap \Gamma_2 \cap \dots \cap \Gamma_m = \phi)$ まず最も確からしい文字集合である Γ_1 に属する候補文字を起点として、文末方向へ a 照合、文頭方向へ b 照合を行う。(a は after, b は before の意で、各照合の仕方は後に述べる。) Γ_1 に属する文字からの照合がすべて

終了すると、照合されずに残った入力文字列について Γ_2 に属する文字から同様の a 照合、b 照合を行う。照合されずに残った入力コラムが存在すれば、さらに $\Gamma_3, \Gamma_4, \dots$ を起点として同様の照合を繰り返す。

DLM-Γ照合法の処理フローを図4に示す。その核心となるアルゴリズムは次のようである。

<< D L M - Γ 照合法 >>

ある Γ 集合に属する文字を c_{i-1} とし、この文字を起点として次の a 照合と b 照合を繰り返す行う。

(1) a 照合:

c_{i-1} の辞書後接 $\{(c_{i-1})^a\}$ と後方入力文字 s_{i+1} の候補文字 $\{c_{i+1}^j \mid 1 \leq j \leq m\}$ とのマッチングを行い、照合するもの ($j = j_a$ で一致するとする) があれば、 $c_{i+1}^{j_a}$ のポインターによる辞書引きを行って次の照合へ進む。

(2) b 照合:

c_{i-1} の辞書前接 $\{(c_{i-1})^b\}$ と前方入力文字 s_{i-1} の候補文字 $\{c_{i-1}^j \mid 1 \leq j \leq m\}$ とのマッチングを行い、照合するもの ($j = j_b$ で一致するとする) があれば、 $c_{i-1}^{j_b}$ のポインターによる辞書引きを行って次の照合へ進む。

1 回の a 照合及び b 照合の照合結果は次の 3 つの状態となる。

< 照合継続状態 >

対象シンボルが検索方向へ接続する場合

< 照合完了状態 >

対象シンボルが語端シンボルの場合

< 照合停止状態 >

対象シンボルが非語端シンボルで検索方向に接続しない場合

照合継続状態である限り照合を続け、同一 Path の a 方向、b 方向の照合結果がともに照合完了状態の場合、その単語が照合されたことになる。ある Path の a 方向、b 方向どちらかでも照合停止状態の場合、単語照合が失敗したことを意味する。

Γ_1 照合が終了すると、入力文字列の中で最も確からしい文字集合に属する文字を単語内に少なくとも一つ含んだ単語の照合が行われたことになる。あるコラムで Γ 照合が成功した場合の処理は次のように行う。

今、 $c_{i-1} \in \Gamma_k$ とし、 Γ_k 照合が c_{i-1}^j ($1 \leq j \leq m$) で成功したとする。

(1) $k < k'$:

新たに c_{i-1} を起点とした照合は行わない。

(2) $k = k'$:

(2a) c_{i-1} が照合された場合

c_{i-1} を起点とした照合は行わない。

(2b) c_{i-1}^j ($2 \leq j \leq m$) が照合された場合

c_{i-1} を起点とした照合を行なう。

また、 Γ_1 から Γ_m の全 Γ 照合が終了しても未照合の入力文字列がある場合、さらに 2 位以下の文字を起点とした照合を行う。

5. D L M 評価

5.1 他手法との比較と照合時間の考察

まえがきのところで述べたように、後処理の手法には文字の接続可能性を用いたものと文頭からの逐次的な文法検定に分けられる。

前者の例に単語確定率^[3]や文字接続表^[6]がある。これらはある文字の前後に接続する単語を 1 次元的に捉えているが、D L M は言語シンボル同志の連関を有向グラフを用いて体系的に捉えたものである。

後者の文頭から逐次的に単語照合と文法検定を行なう最も単純な手法として、奥優先探索の一種がある。これはある時点で尤も妥当な文節パスの選択を行って文節ノードを進めて行き、検定に行き詰まったら前の文節ノードへバックトラックして探索を行うという手法で、文章の区切り(句読点等)に達すると他のパスの探索は行わない。この手法は図 5 のように後方で正解の単語が 1 位に認識されているのに前方からの文節選択の際、正解単語の下位候補文字が前方文節と結びついて誤修正を起こす場合がある。D L M では、確からしい文字から単語照合を行うので、この様な現象は起こらない。

逐次的な他の手法として最良優先探索^{[7][8]}がある。この手法によれば新たに文節を選択する際それまで検索された木の中から最良のものを選択するため前述の現象は起こらない。しかし最良優先探索手法では新たな単語を選択する際、候補文字の第 1 位から第 m 位までの文字を語頭にもつ単語と照合を行っている。この照合法をここでは仮に単語の全照合法とよぶ。D L M - Γ 照合法は確からしい文字を起点とした照合を行なうので、照合時間の短縮をはかるこ

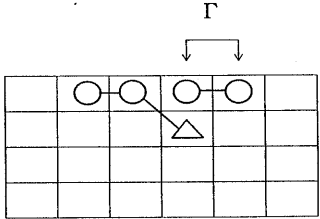


図 5. 奥優先探索での確定単語の "浸食"

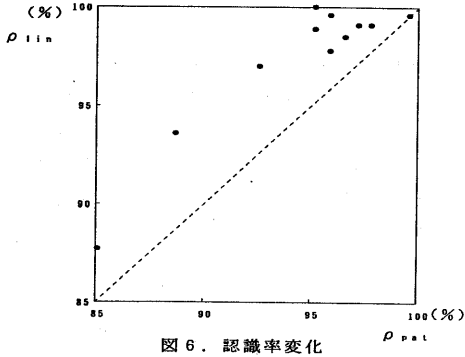


図6. 認識率変化

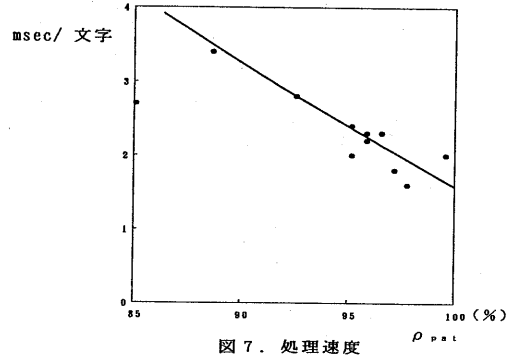


図7. 処理速度

とができる。次に照合時間の考察を行う。

今、認識率 ρ で m 位までの候補文字が得られたとする。単語照合時間は単語長の分布や照合過程から考えて、2文字連接の照合がその主要部分と見なされるので、これにもとづいて粗い試算を行う。ある文字とその隣接文字がともに1位である確率は ρ^2 、1位と2位以下である確率は $2\rho(1-\rho)$ 、2文字とも2位以下である確率は $(1-\rho)^2$ である。また2文字の連接が照合された後のアドレス計算、制御等の負荷を K とすると、DLM- Γ 照合法のおよその単語照合時間は

$$T_D = \rho^2 + 2m\rho(1-\rho) + m^2(1-\rho)^2 + K$$

に比例すると考えられる。 T_D は ρ の2次関数で、 $m > 1$ の時 $0 \leq \rho \leq 1$ で単調減少する。全照合の時間は m^2 なので、最高速の時、処理時間比は $(1+K)/m^2$ となる。

DLMの短所としては、辞書容量が大きいことが挙げられる。辞書サイズは語彙数に左右されるが、PPALは単語羅列型辞書の正味な記憶容量より数倍大きくなる。

5. 2 実験結果

まず、誤読修正評価値を決める。候補1位文字集合を $C^1 = \{c_1^1\}$ 、言語処理前の正解文字集合を R_{pre} 、誤読文字集合を E_{pre} 、処理後の正解文字集合を R_{post} 、誤りが正しくなった文字集合を R_{new} 、処理前は正しかったが誤修正をしてしまった文字集合を E_{new} とする。(以下の集合記号はその要素数を示す。)

パターン認識のみの認識率 ρ_{pre} と言語処理後の認識率 ρ_{post} は

$$\rho_{pre} = R_{pre} / C^1$$

$$\rho_{post} = R_{post} / C^1$$

$$= (R_{pre} - E_{new} + R_{new}) / C^1$$

処理前の誤読文字数に対し、言語処理によって増加する正解文字数の比を

$$\zeta = (R_{new} - E_{new}) / E_{pre}$$

とする。

新聞読み取りデータに対してDLM- Γ 照合法を適用して漢字を含んだ自立語の照合を行った。辞書は学研国語辞書をもとにして、自立語に関するDLM辞書であるPPALを作製した。語彙総数10万で辞書サイズは約4MBである。候補文字の照合深さは $m = 4$ とした。 Γ 照合を行う際は隣接候補文字の上位で一致したPathを優先的に求めた。 Γ 集合は統計的データから求めた誤読されやすさを示す誤読確率をもとにして作製した。シミュレーションでは1位の候補文字を2つの Γ 集合にわけ、1回目は Γ_1 集合を起点として照合を行い、2回目は Γ_2 集合に属する文字を起点とした。2回の Γ 照合で照合されずに残った入力コラムがある場合は2位以下の文字を起点とする照合を行った。使用した計算機はSUN-3/260で4MIPSである。

図6に ρ_{pre} と ρ_{post} の関係を示す。サンプル文書総数は11、漢字を含んだ自立語の全文字数は $C^1 = 5183$ で、評価値の平均は $\rho_{pre} = 95.2\%$ 、 $\rho_{post} = 97.7\%$ 、 $\zeta = 51.4\%$ である。ただし、DLMの性能をみるため切り出しミスの誤読は除いた。誤読修正されていないものは候補もれ文字や1文字単語などであるが、前節で述べた奥優先探索の場合と比べ、表1の様に後方で確定された単語の誤修正が防止されている。現状の Γ 照合法に特有な誤修正として、平仮名を前後の漢字から誤って照合してしまう場合がある。例えば、「三割を国が」→「三

| | 候補文字 | 奥優先探索 | D L M |
|----|------|-------|-------|
| 1位 | 任金 | 任命 | 年金 |
| 下位 | 年命 | ----- | ----- |

(i) 正照合例

| | 候補文字 | 奥優先探索 | D L M |
|----|------|-------|-------|
| 1位 | 中間都市 | 中間部 市 | 中間 都市 |
| 下位 | 市間部東 | ----- | ----- |

(ii) 誤修正防止例

表1. 奥優先探索との比較

(候補文字の下線の付いた文字がΓ₁集合)

割在国」や「字野色を出す」→「字野色速出す」など全体で4件あった。

図7に処理速度を示す。平均で2.3ms/文字である。実線は前節で求めた2次関数でありほぼ予測どおりであるが、認識率85%のサンプルは処理速度がかえって速くなっている。このサンプルでは候補もれ文字が誤読文字の約70%をしめ、他のサンプルより極端に多い。候補もれ文字は連接可能な文字が存在しないことが多くアドレス計算の負荷が減ったため、かえって照合速度が速くなったと思われる。最良優先探索による後処理の処理時間に、1~1.5msec/文字の処理速度が報告されているが^[7]、辞書サイズ、CPU性能を加味すると、DLM-Γ照合法は最良優先探索に比べ1.9~2.9倍高速である。

6. あとがき

本稿では入力文字に対して並列処理が可能な有向グラフ言語モデル(DLM: Digraph Language Model)を提案し、その辞書である前後接リスト(PPAL: Pre-Post Adjacent List)と汎用CPUで処理するのに適した確からしい文字からの照合法であるΓ照合法を紹介した。さらに、DLM-Γ照合法を新聞読取結果に適用した結果、漢字を含む自立語の認識率は95.2%から97.7%になり、照合速度は2.3msec/文字(語彙数10万、4MIPS)であった。DLM-Γ照合法の有効性は確認されたと思う。

誤読修正されていないものの60%は候補もれ文字で、30%は1文字単語である。長さが2以上の単語は候補もれの前後の文字から単語の推測が可能である。その際、語頭の文字が不明で

も、単語の検索ができるDLMは有用である。候補単語の絞りこみや1文字単語の選択は自然言語処理分野における意味論が最適であるが、現状は後処理に適用できるほど、充実していない。しかし、その他の色々な手法を適用する余地はまだあると思われる。一方、誤読修正できなくとも誤った文字を判別する機能も大切であろう。

DLMの短所として辞書サイズが大きいことが挙げられる。EWSをホストマシンとする場合は問題ないが、現状の標準パソコンでは主記憶が不足である。語彙数の削減や、PPALのコンパクト化の検討が必要であるが、記憶デバイスの大容量化は現在も進行中であり、将来的には問題ないと思われる。

今後は、これらの検討を行うとともに、かな部をふくめた接続検定を行う予定である。

[謝辞] 本研究の機会を与えて下さった小室本部長代理、足立部長、並びに辞書作成の準備やその他の援助を頂いた5技)2開)の松井氏始め、課員の皆様に感謝致します。

[参考文献]

- 1) 伊理、他、「グラフ・ネットワーク・マイロイド」、産業図書
- 2) ウィルソン、斉藤他共約、「グラフ理論入門」、近代科学社
- 3) 梅田、「文字認識と単語辞書」、PRU88-17
- 4) 黒沢、「日本語文章を対象とする文字認識後処理方式」、情処理全大、1988
- 5) 紺野、本郷、「日本語OCRのための誤読修正方式の一検討」、情処理全大、1990
- 6) 杉村、他、「文字連接情報を用いた読取り不能文字の判定処理」、電通論誌、Vol. J68-D, No1
- 7) 高尾、西野、「日本語文書リーダ後処理の実現と評価」、情論誌、Vol. 30 NO. 11, 1989
- 8) 長尾、「言語工学」、昭晃堂
- 9) 西野、「日本語文書リーダ後処理の実現」、自然言語処理、1987
- 10) 西原、「整合ラベリング問題と応用」、情報処理、VOL31, No4, 1990
- 11) 丸川、他、「オートマトン型単語照合の高速化手法」、情処、41回全大、1990