

日本語文書リーダ後処理の実現

西野 文人 高尾 哲康

(富士通研究所)

日本語文書リーダ後処理は、文字認識部で認識した結果の候補文字集合列を入力として受け取り、これに対して言語的制約を満たすかどうかを検査することにより正解文字列を推定する機能を持つものである。我々は文書リーダ後処理を探索問題と考え、パターンとしての文字認識による評価値と単語照合検査・単語隣接可能性検査や文字種による隣接確率などの言語的制約によって計算される評価値を用いて、最良優先探索手法により正解文字列を推定する日本語文書リーダ後処理システムを実現した。さらに、評価値の計算方法として確率モデルを導入することによって文字認識の手法に依存せずに言語処理を行なえるようにすることを検討している。本稿では本システムの方式と今後の課題について論じている。

Implementation of Post-Processing for Japanese Document Readers

Fumihito NISHINO Tetsuyasu TAKAO

Software Laboratory, Fujitsu Laboratories Ltd, Kawasaki
1015, Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

This paper describes a post-processing method of the character recognition process, and further development of this method. Given the sequence of candidate character set from the pattern recognition system, this post-processing system searches for the proper sequence by checking it linguistically. This system employs a best-first-search method with the evaluation estimated by the character recognition system and linguistic constraints such as word collating checks, word adjacency checks and character checks. We are now considering a stochastic method, which could be applied to any kind of character recognition method.

1. はじめに

日本語の文書を対象とした文字認識装置（日本語文書リーダ）の必要性が高まっている。日本語文書リーダ後処理は、文字認識部で認識した結果の候補文字集合列を入力として受け取り、これに対して言語的処理を行ない、言語的制約を満たすかどうかを検査することにより文字認識の高精度化を図ろうとするものである。我々は、言語的処理として単語照合検査・単語隣接可能性検査を行ない、さらに、文字種によるヒューリスティック規則や文字の切出し誤りに対処する機能を組み込んだ後処理システムを作成した。本稿では本システムの方式と今後の課題について述べる。

2. 後処理アルゴリズム

2.1 後処理問題の定義

文書リーダ後処理は文字認識から与えられる各文字単位の候補文字集合の列（図1）とその評価値データに対して、初期状態から出発して適当な作用を適用して最終状態のうちのコストを最小にする解を求める探索問題と考えることができる。

文字位置	1	2	3	4	5	6	7	8	9	10	
第1位候補	一般的	在	意志	疎通	の	手段					
第2位候補	メ	股酌	を	章	忘	政	適	〇	乎	殺	
第3位候補	ノ	船	曲	な	煮	吉	珠	逸	卯	手	投

図1 後処理入力データ例

コストは文字認識の評価値データと言語的制約に基づく評価値から計算されるものである。状態空間、初期状態、最終状態、各状態で適用可能な作用は以下のように与えることができる。

1. 状態空間は、次に処理すべき文字の位置と、これまでに処理して確定した単語のリスト（確定単語リスト）の対で与えられる。

（例：（7，〔〈，一般的，な，意志〕）

ここまでの処理の結果の単語リストは〔〈（文頭），一般的，な，意志〕であり、次に7文字めから処理が行なわれることを示す状態）

2. 初期状態は、文字の位置1，文頭記号のみから成るリストを確定単語リストとする状態（1，〔〈〕）で与えられる。

3. 最終状態は、文字の位置（文の最後の位置+1），言語的に正しい単語列を確定単語リストとする状態である。

（例：（12，〔〈，一般的，な，意志，疎通，の，手段〕））

4. 各状態で適用できる作用は、その状態が与える文字の位置から始まる文字認識結果の候補文字集合の列の中から確定単語リストに言語的に接続可能な単語を選ぶことである。この作用を適用して得られる新しい状態は文字の位置を単語の文字数分だけ移動し、選ばれた単語を元の確定単語リストに追加したものである。

（例：状態（4，〔〈，一般的〕）に対して適用できる作用は、単語「在」ないしは「な」を選ぶことで、これにより、それぞれ（5，〔〈，一般的，在〕），（5，〔〈，一般的，な〕）という新しい状態が生成される。なお、単語「を」を選ぶことは、形容動詞「一般的」に格助詞「を」が接続不可能であるという言語的制約により排除される（図2）。）

一般的（形容動詞）←在（地名接頭語）

×を（助詞）⇔接続不可
└な（形容動詞語尾）

図2 後処理の探索

2.2 制御戦略

我々は後処理の探索を最良優先探索(best-first search)により行なっている。以下に探索のアルゴリズムを示す。

procedure 探索

- ① 初期状態Sをopenlistに入れる。
- ② loop
- ③ if openlist = 空リスト then exit(fail)
- ④ n := pop(openlist)
(openlistの先頭要素(最も評価値の高い節点)をnに代入し、openlistからはその要素を取り除く)
- ⑤ if 終了(n) then exit(success)

- ⑥ n を展開し、すべての子節点を生成し、その各々から n へ向かうポイントをつけて、openlist に加え、openlist を評価値の高い順にソートする。

⑦ endloop

すなわち、

①初期状態 S から始めて、

④それまでに生成した節のリスト (openlist) のうちで最も有望そうなものを選び出す。

⑤もし取り出された要素が、文章の終りまたは句読点を認識したならば処理を終了する。句読点を認識しても、まだ文章の終了位置まで達していないならば、その句読点の位置の次の文字から再び探索を始める。すなわち、句読点の認識をもって探索のカットを行なうのである。これは句読点の認識の精度の高さを仮定してのことである。

⑥次にその節の次の文字候補を選び、その文字候補で始まり、候補となり得る単語 (文字認識結果の候補の組合せで作られ、前の単語に接続可能なもの) を作り、評価値を計算することによって新しい節を作り、これまでに生成した節の集合を追加する。

⑦再びこの節の中で最も有望そうな節を選び出しこの過程を続行する。

この探索は基本的には木の探索なので、ループに陥ってしまうことはない。現在のところ、節点の展開時に利用している言語情報は直前の単語の右隣接情報 (右側にどのような単語が接続可能か) だけなので、たとえ確定された単語リストが異なっても、右隣接情報が同じならば同一状態と認定できる。そこで、一度試した状態 (新しい文字の位置と前の単語の右隣接情報の組) はリストとして保存しておき、新たに展開した節点がこのリスト中に含まれないかどうかの検査を行なうことにより、無駄な探索を排除している。

2.3 単語検索

単語照合を行なう方法の1つとして、候補文字集合列からすべての長さのあるいは文字種の変化点で照合の範囲を限定した部分文字列を作成して、そのすべての部分

文字列に対して単語照合を行なう方法がある [1, 2]。しかし、この方法では各文字に対する候補が多くなると辞書参照のためのアクセス回数が多くなる。また、照合範囲を文字種により決定することは「さ迷う」などのひらがな漢字混じりの単語を無視することになってしまう。

我々は、候補文字集合列から部分文字列を作成して辞書を検索するのではなく、辞書から検索してきた単語が候補文字集合列の部分文字列になっているかどうかを検査するという方法を採用した。この方法は次の辞書検索・メンバ検査・接続検定からなる。

(a) 辞書検索

指定された候補文字に対して、その候補文字で始まる単語のすべてを辞書から検索する。

2.1の例では、1文字めの1番めの文字「一」からは「一」、「一円」、「一般」、「一気」、「一揆」、「一般的」、…といった文字「一」で始まる単語がすべて検索されてくる。

(b) メンバ検査

(a)の辞書検索によって引けてきた単語のつづりのそれぞれの文字が、対応する候補文字集合の要素であるかどうか検査する。その単語のつづりのすべての文字が上記条件を満たすような単語を返す。

2.1の例では、「一」で始まる単語リストから「一般的」、「一般」、「一」が残される。

(c) 接続検定

上記で得られた単語が前の単語に隣接可能であるかどうかを検査する。

2.1の例では「一般的」も「一般」も「一」も文頭に隣接可能なので、この3単語が単語検索の結果として返されることになる。

この方法では、単語照合を行なう範囲をあらかじめ制限する必要がない。また、同じ文字で始まる単語をまとめて保存しておくことにより安いコストで辞書検索ができる。1回の検索で引けてくる単語数が問題になるが、7万語の辞書では表1に示すように、ある文字に対して検索される単語の数は高々555個である。また、一般の文章に対しては約1.5文字につき1回の辞書アクセスがあり、1回の辞書検索につき検索される単語数の平均は

69.8単語であった。そして、この中からメンバ検査によって残る単語数の平均は3.8単語（検索される単語数の約5.4%）であった。現在のシステムにおいてこれらの部分の実行時間が全実行時間中に占める割合は図3に示すとおりである。

表1 同一文字で始まる単語数

順位	文字	件数	順位	文字	件数
1	大	555	6	ス	344
2	一	460	7	か	335
3	ア	410	8	不	292
4	小	366	9	中	270
5	お	362	10	無	267

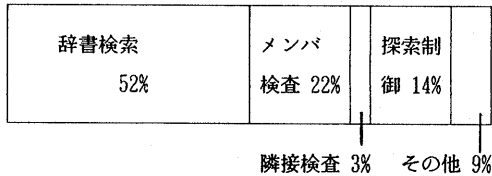


図3 各部分の実行時間中に占める割合

3. 評価値の計算

探索の中から評価値の一番高いものが解として選ばれることになる。この評価値は文字認識部の認識結果による評価値と言語的制約による評価値から計算される。言語的制約による評価値は、単語辞書に単語が存在するかどうかや隣接する単語の接続可能性や文節数によって計算される。さらに、文字種の隣接確率に基づく評価値として、例えば「英小文字が数字に隣接するよりは数字が数字に隣接する確率が高い」とか「英小文字の次に英大文字が隣接するよりは英大文字の次に英大文字が隣接する確率が高い」といった評価値計算を導入することにより、「30u億円」や「cAD」のような認識結果に対して、第2位候補以下に数字「0」や英大文字「C」があれば、それを正解と認識させることを可能としている。

4. 未登録語処理

後処理では与えられた候補文字のすべてを組み合わせ

言語的に正しい文字列を見つけるが、このような全数サーチには次のような時に問題がある。

- ① 辞書にない単語（未登録語）が使用された。
- ② 候補文字の中に正解文字がない。
- ③ 一般的文法にのっとっていない用法の文。

上記のような場合には、どのような候補文字の組合せを試みても正解は得られない。ときとして候補の順位の低い文字を組み合わせて接続的には正しい文字列が出来ることがあるが、このとき本来正解であった文字をも他の文字に変更してしまうかもしれない。これに対処するには、候補文字すべての組合せの全数サーチをやめて、信頼できる文字での後処理を行なうことが必要である。そして、正解文字が候補文字集合にない場合にはその部分を未登録語として認識し、誤りが周囲に拡散するのを防ぎ、正解⇒不正解となってしまったものを救うことが必要である。

未登録語の発見は次のようにして行なう。まず、辞書から単語を引いてきた時に、それらの単語に加えて、未登録語のテンプレートを単語リストに加えておく。未登録語のテンプレートの単語表記の部分には、任意の文字とマッチする文字やカタカナとのみマッチする文字のようなパターンとしての表記記述をしておく。未登録語のテンプレートとしては、2文字からなるサ変名詞のテンプレートや1文字の記号のテンプレートなどを用意しておく。未登録語テンプレートは候補文字集合列と突き合わせを行なうメンバ検査の際にパターンとしてマッチングをとるという以外には一般の単語と同様に取り扱われる。このテンプレートの評価値としては適切な低い値を設定しておけばよい。未登録語用のテンプレートが選ばれた場合には、後処理の最後のステップでその未登録語の処理をする。このステップでは、各文字候補の評価値と文字同士の隣接確率などによって処理される。このような未登録語処理を導入したことにより、1ヶ所の誤認識や未登録語によって周辺部分の後処理に悪影響を及ぼすことがなくなっている。

5. 正解文字が候補文字にない場合の推論

文字認識への入力データ中のよごれやしみなどにより、ある文字に対する認識結果の候補中に正解文字が含まれ

ないこともあり得る。そのような場合には前後のテキスト文から推測してその正解文字を推定することが必要である。

候補文字集合中に正解がない場合の処理としては、助動詞や活用語尾などの単語隣接規則により推定可能な単語と専門用語や外来語のような長いつづりの単語に対して行なわれる。

例えば、「使われる」というような文字認識結果が得られたとすると、人間ならばこれに対する正解は「使われる」あるいは「使わせる」であることがわかる。これは日本語の隣接規則から「札」の部分に来ることができる文字が限られているからである。このような隣接規則から使用される場所が限定されるような助動詞や活用語尾などの単語も登録語テンプレートとして用意しておく。これによって「札」の部分の文字候補中に正解の文字が含まれていなくても救済できることになる（少なくとも他の部分に悪影響を及ぼすことがなくなる）。なお「札」の部分が「れ」か「せ」のどちらかであることの決定は他の言語的制約による評価値と文字認識のこれらの文字に対する評価値によって決定する。

単語のつづりのある文字が候補文字集合中にない場合でも救済できるようにするためには次のようにする。辞書検索後のメンバ検査のところで候補文字集合中に正解の文字がない場合にも、その単語をリジェクトするのではなく単に評価値を下げるようにする。その結果、長いつづりの単語ではある文字が候補文字集合の中に含まれていなくても高い評価値を得ることができる。これによって候補文字集合中に正解を含まない場合でも単語検索ができるので、後処理による救済が可能になる。ただし、単語の先頭の文字に対して候補文字集合中に正解文字がない場合には辞書検索ができないのでこの方法では救済できない。

意味情報の利用や単語の先頭文字における候補文字集合中に正解がない場合に対する対処は今後の課題である。

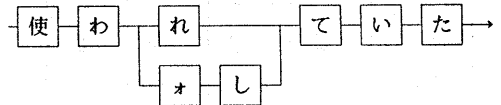
6. 文字の切出し誤りに対する対処

一般に文字の大きさは一定ではないので、文字の切出しを正確に行なうことは難しい。「引」や「働」のような分離文字（1つの文字が左右に2つあるいはそれ以上

に分離される文字）の場合や、「れ」のような本来は分離文字でなくても文字のかすれによって2つの別々の文字に見えてしまう場合などでは、文字の切り出し誤りにより、「引」が「弓1」、「働」が「イ重力」のように2文字や3文字として認識されてしまうこともある。これらに対処するためには言語的情報を利用することが必要である[3]。

我々は文字認識部から別の文字切出しによる候補の可能性を文字開始要素番号と統合要素数として受け取る。ここで、文字開始要素番号は入力データの候補文字の先頭からの文字位置を示し、統合要素数は分離文字の場合の統合された要素の数を示す。例えば、「は」では2、「衛」では3となる。統合された文字の次以降に統合されない場合の認識結果データが続く(図4)。後処理ではそれぞれの可能なパスに対して評価値を計算して評価値の高い候補を選び出している。

例 「使われていた」



探索ルート	1位	2位	3位	4位	文字開始要素番号	統合要素数
→	使	便	疲	伎	333	1
→	わ	わ	ね	め	334	1
→	れ	わ	ね	九	335	2
→	オ	ナ	オ	f	335	1
→	し	L	じ	l	336	1
→	て	で	t	r	337	1
→	い	い	u	U	338	1
→	た	だ	t	允	339	1

図4 文字切出し誤りに対する対処

7. 現在の後処理の能力

以上の方式による日本語文書リーダの後処理を実現した。辞書は現在基本単語約7万語を用意しており、処理速度はM380(15MIPS)で1.5msec/文字を達成することができた。救済能力は後処理前認識率が90%以上であれば70%~100%の救済率(後処理によって救済することができた文字数を後処理前の1位候補の不正解文字数で割ったもの)を達成することができた。表2に現在のシステムで後処理によって救済できたものと救済できなかったものを種類別に見た現象の数(1つの現象に複数の文字の誤りを含む場合もある)を示す。

表2 種類別救済数(現象数)

データ名 文字数 認識率	データ1 516文字 (91% ⇔ 100%)		データ2 639文字 (90% ⇔ 98%)	
	救済	失敗	救済	失敗
単語照合	25	0	30	8
接続検査	4	0	13	0
文字種	5	0	0	0
構文	0	0	0	6
意味	0	0	0	5

データ2において、単語照合で救済できなかったものは、順位上位の候補文字が採用されてしまったもの4件(主に1字漢字+1字漢字)、文字候補集合中に正解の文字がなかったもの3件、単語辞書中に正解のなかったもの1件であった。構文レベルの検査が必要で救済することができなかったものとしては、「役割の大変重要でず」などがある。意味レベルの検査が必要なものは、「~する方法でして(として)…する」、「…社が…社で(と)共同開発した」といったものがあつた。

8. 今後の課題

8.1 確率モデルの導入

文字認識手法に依存しない後処理の評価値の計算法として確率モデルの導入について検討する。

文字認識から後処理への入力として、各文字に対する候補文字集合の他に、認識処理において得られた各文字候補の特徴量・距離値などの評価値を用いることによって、処理の効果をあげることが行なわれている[4]。しかし、この評価値として観測値そのものを使用したのでは、文字認識の手法に依存してしまう。後処理への入力としては、文字認識の手法とは独立な厳密な値が要求される。そこで、Cを本来の文字、Xを観測値(特徴ベクトル)としたとき、各候補文字の距離値を正規化した値とその文字が正解である確率との関係を実験的に求めた評価値を利用する方法を採用しているものもある[1]。しかし、このような結果から元の文字を推測したときの確率 $P(C|X)$ は、Cの分布に強く影響を受ける。したがって、入力の対象となる文字セットの変化などによってCの分布の変化が起こり、この確率 $P(C|X)$ の値は不安定である。

我々は、文字認識から後処理への入力として、現実世界の因果律と同じ方向の確率として本来の文字がCであった時に観測値Xが得られる確率 $P(X|C)$ を用いることを検討している[5]。この確率は文字Cを読む試行を繰り返してその結果得られる観測値の分布を調べることによって安定的に、例えば、 $P(x|”力”)=0.90$, $P(x|”刀”)=0.81$, …のように求めることができる(ここでこれらの合計値が1である必要はないことに注意)。

なお、文献[6]によれば手書き文字認識においては $P(X|C)$ は膨大な数の試料を必要とするので実際には推定不可能であるとしている。しかし、印刷文字の場合には大量の試行は可能であり、この $P(X|C)$ の値は推定可能であろう。また、文字認識と後処理との間のインタフェースとしては、手書き文字であっても印刷文字であっても同一の評価値が望まれる。現時点で $P(X|C)$ の正確な値が得られなくても、ある程度の試料数から判断した近似値は得られるであろう。この確率 $P(X|C)$ の値を精度よく求めることは今後の文字認識側の課題であろう。

文字認識から認識結果を確率として与えられた時に、言語的制約も確率として考えることが必要であろう。単語の生起確率や確率文法を用いた形態素解析の有効性は示されている[7]。ここでは文書リーダ後処理におい

て文字認識と言語的制約における確率の取り扱いを Belief Network [5] として表現することで考えてみる。Belief Networkでは、命題をノードで表現し、2つの命題間の直接的な依存関係をそのノード間を結ぶ原因から結果へ向かう有向アークで示す。有向アークに対しては固定的な確率分布が与えられ、ノードに対しては変数の値の確率分布(確信度)が与えられる。

一般の言語処理を施した文書リーダの後処理のモデルとしては図5のようなものが考えられる。

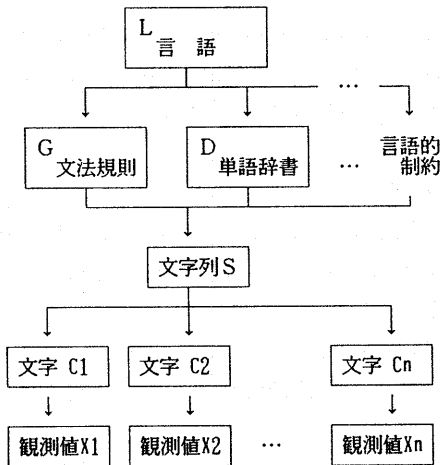


図5 文書リーダ後処理のBelief Networkモデル

ノードLは言語(および文書の形式・文体)に対する確信度を示すものであり、入力文書の言語が日本語であるとか、中国語であるとか、単なる意味のない文字のつながりであるとかいったことに関する確信度を示すものである。この言語や文体に関する確信度に従って、その言語における文法に関する確信度や、使用される単語に関する確信度などの様々な言語的制約に関する確信度が得られる。これらの言語的制約による確信度に従って入力文字列Sの確信度が得られる。この入力の文字列Sに対する確信度は、文字認識における観測値が得られる前では、その言語におけるそれぞれの文字列が出現する頻度に基づく値が与えられるであろう。今、文字認識によって各文字に対する観測値 X_1, X_2, \dots, X_n が得られたとしよう。この観測値に基づいて文字列Sにおける各文字 C_1, C_2, \dots, C_n に対する確信度が得られ、これによって文字

列Sにおける各文字列の確信度が変化することになる。この文字列Sにおける確信度の変化は、辞書Dにおける確信度の変化(未登録語の認識に相当する)や言語Lに対する確信度の変化を引き起すこともある。このようにして、それぞれのノードに対する確信度が計算されていく。後処理結果は文字列Sにおける確信度の最も高い文字列として与えられることになる。

このモデルにおいては、単語辞書は固定的な確率分布 $P(D|L)$ に相当する。すなわち、つづり D_w に相当する単語が存在することは $P(D_w|L) = \delta (>0)$ として表現され、つづり D_x に対する単語が存在しないことは $P(D_x|L) = 0$ として表現されることになる。 δ の値としては言語Lにおける単語 D_w の頻度によって変化させることができる。このようにすることによって、辞書の単語数が増加しても、本来正解ではなかったものがたまたま辞書に存在していたために選択されてしまうことがおきてしまうことをある程度防ぐことができる。

現在の段階では、文字認識の確率 $P(X|C)$ も言語的制約の確率 $P(G|L)$ や $P(D|L)$ も正確に求めることは困難かもしれない。しかし、このような確率の考え方を導入して進めていくことは、①文字認識と言語的処理とを独立に開発することができ、②理論的根拠に基づいて開発者・利用者がヒューリスティックス規則などをチューニングすることが可能である、という利点がある。今後このような確率モデルを導入することは必要なことである。

8.2 様々な種類の文書への適用

現在の後処理システムでは、言語的制約に基づいて評価値を計算して処理を行なっている。そこでは日本語の文法や単語としてはある程度標準的なものを仮定している。機械翻訳のようなシステムでは、入力される文章の種類としてある程度の規約を設けることは可能であった。しかし、文書リーダにおいてはこのような入力文書に対する制限を設けることは好ましくない。一般の文書に対する救済率を下げることなしに文法的な制限の緩和と辞書的な制限の緩和を行なう必要がある。

8.2.1 単語隣接可能性検査

一般の助動詞などは前に来る単語の品詞の種類や活用

形が限定されているので、厳密な接続検査が可能である。しかし、名詞+動詞のような関係は、一般の文章では接続不可であるが、新聞の見出しなどの特定の分野では接続することもありえる。この接続可能性を0か1ではなく0と1の間の値をとる確率として表現するとともに、応用分野によって切り替えが可能にすることが必要である。

8.2.2 単語表記のゆれ

機械翻訳のようなシステムでは単語を一定の表記で表現するという規約を設けることは可能であった。しかし、文書リーダではこのような入力文書に制限を設けることは好ましくない。日本語の文書中には、「組合せ」と「組み合せ」のような送り仮名のゆれ、「くみあわせ」や「組みあわせ」などのようなひらがなによる表記、「パラメタ」と「パラメータ」のような外来語カタカナ表記のゆれ、「長島」と「長嶋」のような異字体などのさまざまな表記のゆれがある。文書リーダとしては、これらのどの表記がでてきても受理するようにしなければならない。これらの問題に対処する方法としては、すべての可能性を辞書に登録する方法とプログラムで処理する方法が考えられる。またすべてカタカナの文章などの特殊な場合にはあらかじめ別の辞書を作成する方法も考えらる。これらの問題に対処するのは今後の課題である。

8.3 高精度化

人間が文字認識の後処理を行なった場合の精度と利用している知識については[9]で報告した。救済率の高精度化及び低認識率に対する救済率の向上のために[9]で述べたような語と語の関係情報の利用、関連語情報の利用、文脈処理、文書構成の知識の利用などを実現することは今後の課題である。

9. むすび

本稿では日本語文書リーダにおける言語的処理を行なう後処理の方式と今後の課題について述べた。実際に文書リーダを使用する場合には文章が正確に読めたかどうか利用者のためには問題である。完全に100%の認識率を得ることができない現在では、文書を読み込んだ後の

修正のしかたについても十分考えることが必要である。本稿で述べたような確信度にもとづく後処理では、各文字の確信度に応じて人間への問合せをするなどのインタフェースが作成できるであろう。また、利用者の入力文書の文体や単語の特性による後処理システムのチューニングを可能にすることも必要であろう。人間とのインタフェースを考えた後処理システムを作成していくことが大切であろう。

謝辞

本研究に対して貴重な助言をいただきました富士通研究所の山本栄一郎主任研究員、武理一郎氏、ソフト1研の研究員の方々に感謝いたします。

参考文献

- [1] 池田, 大田, 上野: 「手書き原稿認識における語彙および構文の検定」, 情報処理学会論文誌 Vol. 26, No.5 (1985)
- [2] 新谷, 目黒, 梅田: 「言語情報と認識情報の複合利用による文字認識後処理」, 研究実用化報告 Vol. 34 No.1 (1985)
- [3] 村瀬, 新谷, 若原, 小高: 「言語情報を利用した手書き文字列からの文字切出しと認識」, 電子通信学会論文誌 Vol. J69-D No.9 (1986)
- [4] 新谷, 目黒, 池田: 「言語情報と認識情報を用いた文字認識後処理」, 電子通信学会技術研究報告 PRL82-76 (1982)
- [5] J. Pearl: "Fusion, Propagation, and Structuring in Belief Networks", Artificial Intelligence 29 (1986)
- [6] 長田, 日高: 「一括処理法による手書き文字の認識」, 電子通信学会論文誌 Vol. J66-D No.2 (1983)
- [7] 松延, 日高, 吉田: 「確率文節文法による形態素解析実験について」, 九大工学集報 Vol. 59 No.6 (1986)
- [8] 長田, 牧野, 日高: 「日本語の文脈情報を用いた文字認識」, 電子通信学会論文誌 Vol. J67-DNo.4 (1983)
- [9] 西野: 「文字認識後処理の可能性」, 自然言語処理研究会資料62-10 (1987)