

## 文字タグ法による手書き住所読み取りの評価

4R-7

福島 俊一† 下村 秀樹† 森 義和‡

(NEC† NEC情報システムズ‡)

### 1 はじめに

郵便物の宛名住所のようにフリーピッチで書かれた手書き文字列は、字形が多様で、文字サイズにばらつきがあり、文字の接触・入り組みなどもよく起きる。したがって、その高精度な読み取りの実現には、誤切り出しや誤認識によって欠落した正解文字を補完する知識処理(文脈処理や後処理とも呼ばれる)[1]が不可欠である。

現在主流となっている知識処理方式は、単語辞書と認識結果文字列(文字切り出し/文字認識の多候補)を照合して単語候補を抽出した上で、その並びの妥当性を判定する2段構成である。このような従来法は、単語境界が不確定なケースをうまく扱えない、単語辞書との虫食い照合における組み合わせ爆発を避けると強引に可能性を切り捨てざるを得ない、などの問題を抱えている。

それに対して筆者らは、新しい知識処理の枠組みとして文字タグ法を提案した[2]。文字タグ法は、文字を基本単位として動的計画法による最良解探索を実行するので、組み合わせ爆発を回避し、効率性と精度を保証する。本稿では、文字タグ法を手書き宛名住所の地名領域の読み取りに適用して評価し、その有効性を確認する。

### 2 文字タグ法

一般に読み取りで用いる知識は、読み取り対象文字列に出現し得る単語のリストと、それらの単語間の接続可否情報であり、図1のような単語テーブルで記述できる。図1は、読み取り対象文字列に「川崎市」「宮前区」「梶ヶ谷」「宮崎」…「有馬」が出現し得て、「宮前区」には「川崎市」、「梶ヶ谷」「宮崎」…「有馬」には「宮前区」が前接可能であることを表わしている。

さらに、文字タグ法では、単語テーブルから機械的な変換によって作成した文字インデックスを参照する。図2は、図1の単語テーブルから作成した文字インデックスであり、例えば「崎」という文字は「川崎市」の3文字中の2文字目、あるいは「宮崎」の2文字中の2文字目になり得ることを表わしている。

図3は文字タグ法による処理例である。まず、先頭側のセグメントから順に、すべてのセグメントのすべての候補文字について、文字インデックスを検索して得られる3項情報をもとに文字タグを生成する。次に、先頭側の文字タグから順に、すべての文字タグについて、それより先頭側の文字タグで、連結可能な位置関係の条件を満たし、連結したときのコストが最良になるものを見つけて連結していく。

文字タグの連結時コストは、前側の文字タグのコストに後側の文字タグの初期コストと2つの文字タグ間の間隔にもとづくコストとを累積したものである。ただし、

0:	["川崎市",	{ }	4:	["宮前平",	{ 1 }
1:	["宮前区",	{ 0 }	5:	["馬絹",	{ 1 }
2:	["梶ヶ谷",	{ 1 }	6:	["野川",	{ 1 }
3:	["宮崎",	{ 1 }	7:	["有馬",	{ 1 }

図1: 単語テーブル

ヶ:	[2,2,3]	川:	[0,1,3]	[6,2,2]
梶:	[2,1,3]	前:	[1,2,3]	[4,2,3]
宮:	[1,1,3]	谷:	[2,3,3]	
区:	[1,3,3]	馬:	[5,1,2]	[7,2,2]
絹:	[5,2,2]	平:	[4,3,3]	
崎:	[0,2,3]	野:	[6,1,2]	
市:	[0,3,3]	有:	[7,1,2]	

図2: 文字インデックス

連結時コストが極端に悪くなる場合は文字タグの連結をカットする。また、単語列の先頭/末尾の読み飛ばし分についても補正コストを加える。動的計画法によるコスト計算と文字タグ連結により、最良コストとなった文字タグ連鎖を最終的な読み取り結果とする。

アルゴリズムの正確な記述は前稿[2]を参照願いたい。 $L$ を入力データの文字列長、 $M$ を正解文字当たりの平均候補文字数としたとき、本アルゴリズムの時間計算量は $O(L^2 \cdot M^2)$ (ただし、通常のコスト定義では $O(L \cdot M^2)$ )、メモリ使用量は $O(L \cdot M)$ である。

### 3 評価

評価では、葉書に書かれた縦書きの手書き宛名住所データを用い、石寺らの方式[3]による文字切り出しと、津雲の多段弾性照合法[4]による個別文字認識の実行結果を、文字タグ法の入力とした。正解文字数に対するセグメント候補数の比は平均1.9個、各セグメント当たりの候補文字数は平均2.8個(最大5、最小1)で、 $M = 1.9 \times 2.8 = 5.3$ である。

読み取り対象の地名は、東京都国分寺市の全域(セットK)と、静岡県富士市の全域(セットF)の2種類を用意した。各セットの読み取りのために単語テーブルに登録した住所要素(都道府県名・市区郡名・町名など)の規模は、セットK: 49個、セットF: 231個である。

文字タグ法の読み取り精度は、正解地名文字列の構成文字が入力データ中にどれくらい含まれていたかの割合 $R$ (正解文字存在率)と関係付けて測定した。図4に示した $R$ の値域ごとに50件ずつのデータをピックアップして測定した。文字タグ法の第1候補が地名領域の正解文字列と完全一致したかどうかで判定した。第1候補が不正解となった原因には次のようなものがあり、その比率は図4に示した。

- a: ある住所要素の全文字が入力データ中がない。
- b: 誤認識文字が偶然に別な地名候補に読める。
- c: 長く伸びたストロークがノイズセグメント化。
- d: 文字タグ連結カット条件により市区郡名が欠落。

Evaluation of a Post-Processing Algorithm for Hand-Written Address Reading  
Toshikazu Fukushima†, Hideki Shimomura† and Yoshikazu Mori‡ (NEC Corporation† and NEC Informatec Systems Ltd.‡)

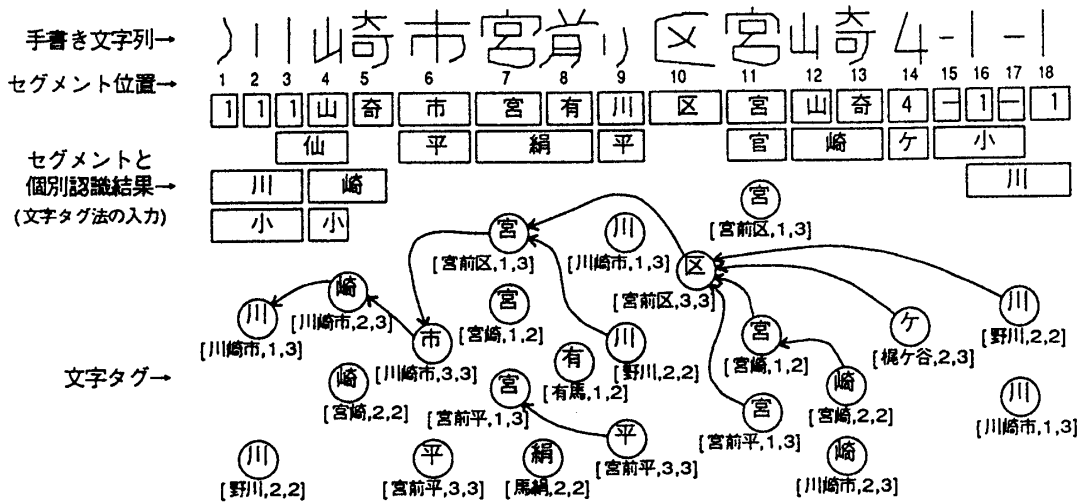


図 3: 文字タグ法による処理例

正解文字存在率 $R$	セット K		セット F		不正解の原因							合計	
	1位正解	不正解	1位正解	不正解	a	b	c	d	e	f	g		
$0\% \leq R < 40\%$	5	45	4	46	69	22							91
$40\% \leq R < 60\%$	21	29	13	37	11	36			19				66
$60\% \leq R < 80\%$	43	7	34	16	7	3	4	2	2	3	2		23
$80\% \leq R < 100\%$	44	6	47	3		3	1		3	1	1		9
$R = 100\%$	50	0	47	3						3			3
合計	163	87	145	105	87	64	5	2	24	7	3		192

図 4: 文字タグ法の読み取り精度の不正解の内訳

- e: 微妙なコスト差による。
- f: 番地領域と部分一致する長い町名が存在。
- g: 同一町名の異表記の方が一致度が高くなった。

処理速度は、NEC EWS4800/330 (CPU: R4400、クロック:67MHz、主記憶:64MB) で 20 ミリ秒未満 / 件であった。文字タグの生成数は、セット K で平均 76.7 個 / 件、セット F で平均 217.5 個 / 件となった。

#### 4 考察

時間計算量から明らかではあったが、虫食い照合を含めて文字切り出し / 文字認識の多候補のあらゆる組み合わせを高速に探索できる文字タグ法の能力が、実データでの評価結果により改めて確認できた。

評価に用いたデータの多候補度は  $M = 5.3$  であるから、文字列長が 8 の場合を単純な目安として取り上げると  $5.3^8 = 60$  万通りの組み合わせのなかから最良なものを探ることになる。図 4 によれば、入力データが上述のような多候補度をもっていたとしても、そのなかに正解文字数の 8 割以上が含まれていれば、文字タグ法の第 1 候補正解率は 94% になる。

図 4 で第 1 候補が不正解になったものでも、原因分類で e~g に該当するケースでは、第 1 候補の読み取り結果と僅差のコスト値をもつ別候補のなかに正解の読み取り結果が含まれている。この点を考慮して、正解文字の欠落(虫食い)がある不完全な入力データから可能性の高い読み取り候補(複数通り)を推定する能力という観点でみると、入力データ中に正解文字数の 6 割以上が含

まれていれば 90% 以上のケースにおいて文字タグ法は有効な探索結果を出力している。

図 4 の原因 a・b のケースは、正解文字がほとんど含まれないのであるから、探索効率の問題ではなく、棄却能力の問題である。

#### 5 おわりに

文字タグ法を手書き宛名住所の地名領域の読み取りに適用して評価し、その有効性(効率性と精度)を確認した。現在の文字タグ法は、常に最良コストのものを読み取り結果として出力し、信頼性の低い場合に読み取り結果を棄却する仕組みをまだそなえていない。今後は、コスト差を分析したり、トップダウン的な検証 [5] を組み合わせるなどして、文字タグ法に棄却判定能力をもたせていくことが課題と考えている。

最後に、評価に協力いただき多くの助言をいただいた山内俊史氏をはじめとする NEC 関係者に深謝する。

#### 参考文献

- [1] 西野、文字認識における自然言語処理、情報処理:34(10)、1993 年。
- [2] 福島ほか、手書き住所読取りのための町名検索アルゴリズム-文字タグ法-、情処 50 全大:4D-6、1995 年。
- [3] 石寺ほか、手書き住所読取りのための文字切り出し方法、信学総大:D-576、1995 年。
- [4] 津雲、手書き漢字認識、NEC 技報:47(8)、1994 年。
- [5] 下村ほか、手書き住所読取りにおけるボタン処理と連携した住所知識処理方式、情処 50 全大:4D-1、1995 年。