自 然 言 語 処 理 82－3
（1991 3. 15）

住所文字列に対する文字認識後処理方式の検討

磯山秀幸

NTT データ通信株式会社

OCR による文字認識結果に対して、誤読文字の検出・修正を行なうために、文字認識後処理を行なうことが試みられている。本研究では対象分野を住所に限定して、住所文字列特有の性質を利用する方法を検討し、評価用システムを作成した。実際の認識結果データを用いて評価を行なった結果、住所文字列の中で地名部分については、実用的な精度が得られ、番地部についても実用化への見通しが得られた。本稿では住所表記構造の解析を中心に単語照合・構文解析の考え方を応用した処理アルゴリズムと試作システムの評価結果を報告する。

POST-PROCESSING FOR CORRECTING THE CHARACTER RECOGNITION OUTPUT TOWARD ADDRESS

Hideyuki Isoyama

NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

Post-processing methods for character recognition output have been researched to detect and correct errors which appear in OCR output. In our research the item written on OCR forms is narrowed down to addresses and the special characteristics in the way an address is written are made use of. From the evaluation by a prototype system, we have proved the effectiveness to the part of an address defined in a dictionary, and clarified the methods for the improvement to the rest of it. This paper describes the algorithms about the word matching and applied syntactic analysis as well as the evaluation results.

1 はじめに

帳票等のデータ入力の自動化の方法として、従来からOCR(光学式文字読み取り装置)が利用されてきた。しかし現在の文字認識技術をもってしても、100%の精度で帳票の文字を読みとることは出来ない。そのため帳票の内容を、正確にコンピュータに入力するためには、人間が読み取り結果の確認と訂正を行なわなければならない。文字認識後処理ではこの負担を最小限に抑えることを目的として、文字認識結果の「リジェクト文字(読み取り不能文字)の救済」と「誤読文字の検出・修正」を行なう。これまでに、文字認識後処理の手段として入力文字列の性質を利用し後処理を行なう方法が試みられている[1][2]。本研究では対象分野を住所に限定して、住所文字列特有の性質を利用する方法を検討し、評価用システムを作成した。本稿では文字認識の後処理として住所表記構造の解析を中心に単語照合・構文解析の考え方を応用した処理アルゴリズムと試作システムの評価結果を報告する。

文字認識の対象 一般に認識の対象は印刷文字と手書き文字に分けられ、また手書き文字の中では「一文字ごとのマス目あり(固定ピッチ)の帳票」と、「マス目なし(フリーピッチ)の帳票」に分けることができる。文字ごとのマス目がない帳票の場合は、一文字ごとを認識処理する前の段階で、文字切り出しを行なうが、ここで誤りが発生する可能性がある。このため一般に、手書きのフリーピッチ帳票の認識精度が最も低くなる。本研究では「文字ごとのマス目ありの帳票に手書きで書かれた住所文字列」を対象とした。ただし、帳票には「都道府県・市区郡・町村」や「丁目・番地」などのフィールドごとの区切りはない。

第1位	東	都	庶	東	都	市	下	烹	区	葛	龍	渥	町
第2位	京	郡	府	東	郁	申	不	京	医	幕	籍	屋	財
第3位	烹	郁	狩	京	郡	帝	木	衰	匡	篇	簾	屈	酎
第4位	茨	部	廓	菓	那	京	市	宗	匠	薦	龍	届	貯
第5位	束	廓	廊	巣	部	牢	末	哀	匿	蒐	簸	厘	鮒
第6位	哀	那	柿	某	郵	宇	干	菜	匝	嘉	薩	居	駒
第7位	文	郵	唐	慕	郭	卒	本	淀	匹	蕩	篭	座	斬
第8位	大	郭	県	葉	郎	辛	ホ	菟	返	謇	薙	産	折
第9位	奈	良	庇	棄	静	柿	F	荒	臣	島	纏	屠	冊
第10位	涙	郤	荷	菜	群	中	ァ	意	匪	蔦	範	雇	針

『京都府京都市下京区葛寵屋町』と書かれていた帳票を文字認識した結果

図 1: 認識結果文字列

認識結果文字列 今回の評価実験で用いたOCRは、標準では一つの文字パターンに対して、図1のように10個の候補文字を出力する。この順位は、各々の候補文字に付与されている距離値[1]と呼ばれる値の昇順に並べられている。この例では、帳票の第1文字目に『京』と書かれていた文字パターンを文字認識した結果、第1位「東」から第10位「涙」までの10個の候補文字が出力されたことを表している。上位にある候補文字ほど文字認識の結果が確からしいものである。図1のようなデータを認識結果文字列と呼ぶことにする。

2 住所文字列の文字認識結果への適用

住所文字列の各部分の名称を図2のように定義する。「都道府県・市区郡・町村」の部分を地名部と呼び、その後に続く「丁目・番地」以降を番地部と呼ぶ。更に番地部を番地フィールド・方書(かたがき)フィールド・部屋番号フィールドの3つに分ける。

2.1 地名部の処理

地名辞書とインデックステーブル 地名部解析のプログラムで使用する単語の辞書としては、市販の「コマキの郵便番号辞書」を利用し、テーブル形式の地名辞書を作成した。この辞書は都道府県・市区郡・町村の各レベルごとに地名単語が格納されており、階層構造をなしている。この辞書を用いると、上位レベルの地名が決定すれば、次レベル

[1]距離値とは文字パターンの適合度合いの逆数であり、値が大きくなるほど確からしさは低くなる。

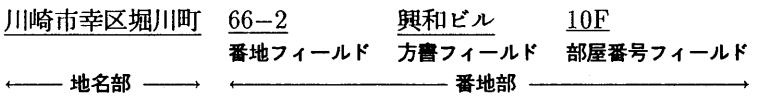


図 2: 住所文字列の各部分の名称

は検索範囲を限定して検索することができる。また、逆に下位レベルの地名が決定されれば、それを含む上位レベルの地名を取得することができる。

この地名辞書を基にして、各文字位置ごとに文字コードでソートしたテーブルを作成した。このテーブルをインデックステーブルと呼ぶ。地名辞書とインデックステーブルのイメージを図3に示す。地名辞書内 (図3では市区郡名の辞書だけを示している。) の単語にはそれぞれ一意にレコード番号が付与されている。図3において各テーブルの数値は、地名辞書内の各単語のレコード番号を表している。インデックステーブルを用いた地名辞書のアクセス方法については、後で述べる。

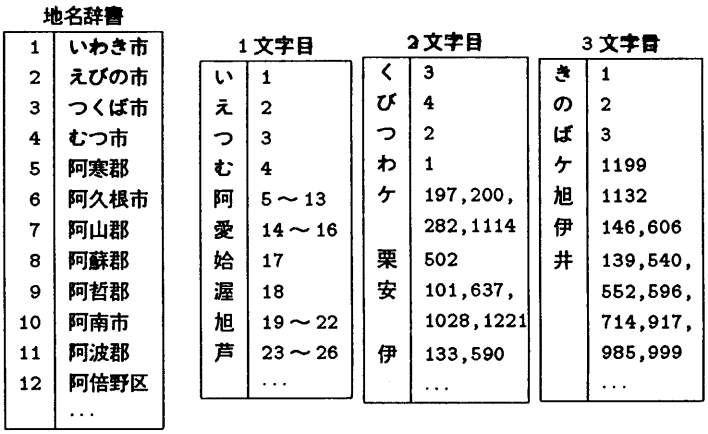


図 3: 地名辞書とインデックステーブル

郵便番号による照合　帳票の郵便番号欄は数字だけが記入されていると考えられるため、パターン認識の辞書を数字のみに限定して読みとることができる。よって郵便番号の文字認識精度は高くなる。帳票に郵便番号が記入されている場合は、郵便番号から検索して得られた住所文字列と認識結果文字列との照合を行ない、これで既定値以上の類似度が得られれば、地名部文字列として優先的に採用する。しかし、郵便番号が書かれていなかったり、または書かれていても照合の結果、誤りであると判断した場合には、以下に示すように、住所文字列の構造を利用した照合によって認識結果の訂正を行なう。

住所構造知識の定義　図4に示すように照合範囲切り出しのキーとなる文字 (県, 市, 区, 町など、以後「キー文字」と呼ぶ) を用いて、住所文字列を都道府県・市区郡・町村などのレベルに分ける。そして属性情報として、切り出しキー文字の直前に現れる文字種と文字数を定義する。

たとえば、"2:5HJ? 市" は、'市' の前に入る文字が2文字以上5文字以下 "2:5"、文字種が「ひらがな "H"」または「漢字 "J"」であり、この範囲が省略可能 "?" であることを意味する。

地名部処理 (住所構造解析)　複数の候補文字を含んだ文字認識結果文字列において、住所構造知識に定義された単語切り出しキーとなる文字をサーチし、定義された住所構造に一致するようにキー文字の組合せを求める。こうして得られた組合せを住所構造の候補とする。実行例として、『千葉県市原市光風台５－６－６－１１』と書かれていたものを、読み取った結果と、その認識結果から作成した住所構造候補を図5に示す。この例ではキー文字として、3カラム目の [県] と、4,6カラム目の [市] が抽出され、住所構造の候補が2つ作成されている。

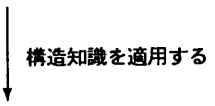
```
((2:2j? 都
  (2:4hj? 市
   (1:5HJ 町 1:5HJ 村 1:6HKJ)
   1:3J? 区
   (1:5HJ 町 1:5HJ 村 1:6HKJ)
   3:3J? 郡
   (0:0J? 大 (0:0J? 字 (1:5HJ 町 1:5HJ 村 1:6HKJ))))
  ...
(1:2N? 丁 (0:0? 目 (1:4N? 番 (0:0? 地 (1:4N? 号 1:4N?-) 1:4N?)))
 1:4N- (1:4N?- (1:4N?- (1:4N?))))

    ?: 省略可能  N: 数字  H: ひらがな
    K: カタカナ  J: 漢字  A: アルファベット
```

図 4: 住所構造定義ファイルの内容

認識結果文字列と地名辞書との照合を行なう場合、認識結果文字列の一部分を切り出して照合を行なうことになる。本方式では、あらかじめ認識結果文字列の切り出し位置と、その位置に対応する地名のレベルを決めてから単語照合を行う。

第1位	千	葉	県	帝	原	市	光	風	合	5	ー	6	ー	6	ー	ノ	ノ
第2位	ヂ	襲	具	市	源	帝	洗	嵐	台	ぢ	ー	占	ー	る	ー	メ	／
第3位	チ	集	鼻	京	廉	牢	濱	夙	今	ぎ	一	ら	一	占	一	／	メ
第4位	千	薬	呉	宇	厚	申	先	鳳	含	互	三	ム	＝	5	＝	ィ	イ
第5位	乎	藁	貝	涌	尿	宇	沈	鼠	自	字	＝	呂	と	百	三	汐	ソ
第6位	辛	棄	臭	滞	屈	苧	流	蚊	分	烹	ニ	5	〔	舌	と	ソ	ォ
第7位	才	糞	曇	希	康	中	兆	厭	舎	占	二	淫	「	召	ニ	ブ	%
第8位	斗	棄	崇	幸	厘	守	滝	凧	谷	享	〜	右	ど	台	ら	ク	汐
第9位	平	藻	員	牢	展	帯	老	寂	命	よ	と	左	ヽ	白	〜	%	”
第10位	子	隻	憂	す	度	亭	発	臥	昏	夕	ら	這	三	ち	二	グ	ゾ



住所構造の候補

1. ○○県○○市○○○

2. ○○○市○○○○○

図 5: 住所構造候補の作成

辞書照合の方式　認識結果文字列と地名辞書とのマッチングをとる際の照合方式を、図5を用いて示す。図5において地名部となるのは、第9カラム目「千葉県市原市光風台」までの部分である。

はじめに、第1カラム目に対する認識結果の第1位候補である“千”をキーとして、都道府県レベルの地名単語の第1文字目でソートされたインデックステーブルを検索し、第1文字目が“千”である地名単語のレコード番号を得る。次に第2位候補である“ヂ”で同様に検索を行なうがこの場合は、該当する地名は存在しない。以下、候補順位の最後までこの処理を行ない、第1カラム目に対する処理を終える。第2カラム目以後についても上記の処理を行ない、このレベルの切り出し文字幅である第3カラム目まで処理を続ける。こうして候補となる地名単語のレコード番号群を得る。このレコード番号群をマージ・ソートし、単語長の半数回以上現れたレコード番号で表される地名は、類似性があると判断して、これらを抽出する。この例では抽出されるレコード番号は、「千葉県」を表すレコー

ド番号のみである。よってこの時点で「千葉県」に決定することができる。次に第4カラム目から第6カラム目までを市区郡レベルとして、地名辞書との照合を行なう。この場合、上位レベルが「千葉県」として決定されているため、地名辞書の階層構造を利用して、千葉県下の市区郡だけに絞って処理を行なう。さらに次の町村レベルについても、同様にして地名辞書との照合を続けていく。

レコード番号のマージ・ソートが終了した時点で、複数の地名が候補として残っている場合は、認識結果文字の距離値を元に評価値を計算して、最も確からしいものを選択する。この時、「一致した文字数が最大のものの中で、評価値が最小であるもの」を正解として選択する。評価値の計算方法を以下に示す。

評価値 = 一致した候補文字の距離差の和
距離差 = 第 n 位候補の距離値 − 第1位候補の距離値
(認識結果の第1位候補文字の場合は、距離差は0になる)

2.2 番地部の処理

地名部の解析が終了した位置より、以後を番地部と考える。図2で示した番地部の各フィールドの照合方法を以下に示す。

番地フィールドの照合　番地部の番地フィールドでは、地名部と同様のルールを用いてキーとなる文字に着目した処理を行う。番地フィールドでは、キー文字以外は全て数字であると考えて、数字を優先的に採用する処理を行なう。この処理では辞書照合は行なわない。代りにキー文字とキー文字との間には必ず数字が入るため、候補文字の中に含まれている数字を優先的に採用する。

部屋番号フィールドの照合　次に述べる方書フィールドの終端を決定するために、先に部屋番号フィールドの照合を行なう。部屋番号フィールドはその開始位置が未確定であるため、住所文字列の終端から先頭方向に向かって処理を行う。処理内容は番地フィールドと同様で、数字および部屋番号フィールドによく表れる英字(例えば「5f」など)を優先的に採用する。

方書フィールドの照合　番地・部屋番号フィールドとして解析不能であった部分については、ビル名・アパート名等の方書であると考えられる。この部分に対しては方書専用の辞書を用いて照合を行い、類似の単語があれば、これを採用する。方書辞書の内容は、処理している住所文字列の地名部で使われた単語と、方書フィールド用として定義された「マンション」「コーポ」「様方」などの接頭語・接尾語的に使われる単語23語である。

2.3 期待できる効果

本システムで用いている方式によって、期待できる効果を述べる。

1. 住所構造知識を基に認識結果文字列を解析することで、前もって無効な候補文字の組合せを、除外することができる。さらに住所構造を外部ファイルとしてまとめて定義できるため、住所構造知識の追加・変更が容易となる。

2. あらかじめ、認識結果文字列の切り出し位置と照合する地名のレベルを決めてから単語照合を行うことで、候補文字の組合せ数を減少できる。また地名辞書の階層構造を利用して単語の検索範囲を制限し、処理時間の短縮と修正誤りの減少を目指している。

3. 認識結果文字列と地名単語との照合処理においては、認識結果の中に正解文字が含まれていないことがあるため、「完全に一致する」単語ではなく、「最も類似した」単語を検索しなければならない。

   認識結果文字列は2次元のマトリックスになっているため、これを組合せて1次元の文字列を作成し、地名辞書から類似した単語を検索したのでは、比較回数が膨大になってしまう。本方式ではインデックステーブルを用いているため、類似単語を選び出すのに、「候補文字数(1文字パターンあたり10文字) * 単語長」回だけの文字比較で済む。よって、組合せ文字列を比較する方式に比べ、総比較回数が少なく高速化が図れるため、類似単語を検索するのに有効である。

3 評価実験と考察

実際にOCRから出力されたデータを用いて、後処理の試作プログラムの評価実験を行なった。住所は県名などを省略した表記で書かれることがあるため、評価実験用の入力データには、以下のようなパターンを用意した。

- 帳票に都道府県名から書き始められた住所 (47 各都道府県)
- 政令指定都市名から始められた住所 (11 政令指定都市)
- 区名から始められた住所 (11 政令指定都市, 東京 23 区)
- 町村名から始められた住所 (任意)

評価実験では、「手書き・マス目あり」で書かれた帳票を、OCRで文字認識した処理結果を用いた。動作環境を表1に示す。

表 1: 動作環境

処理マシン	CPU:68030, UNIX systemV
記述言語	C 言語
地名辞書	コマキの郵便番号辞書
辞書サイズ	約 1.4(Mbyte)
入力帳票	手書き・文字ごとのマス目あり

3.1 評価実験の結果

評価結果を表2に示する。今回の評価で用いたデータは、OCRによる画像パターン認識のみ (OCR独自の単語照合なし) の処理結果である。このため文字認識後処理を行なう前では、住所文字列単位で14.8%、文字単位で88.8%の認識精度しかないデータであった。このデータを用いて試作システムで処理を行なった結果、地名部文字列では91.5%の救済率を得ることができたが、番地部文字列の救済率は46.0%にとどまった。番地部では認識誤りを訂正できたのは、90%以上が番地フィールドであった。これは、方書および部屋番号を含んだ住所が記入された帳票が少なかったことと、方書フィールドの処理精度が悪いためである。

表 2: 実験結果

	地名部		番地部		住所全体	
	文字列単位	文字単位	文字列単位	文字単位	文字列単位	文字単位
データ件数	689 住所	6232 文字	689	4761	689	10993
未処理時正読率	52.1%	92.5%	28.3%	80.9%	14.8%	88.8%
処理後の正読率	94.2%	99.4%	57.3%	87.9%	57.2%	94.4%
救済率	91.5%	91.7%	46.0%	36.7%	51.8%	50.0%
誤訂正率	3.1%	0.3%	6.7%	0.4%	23.5%	0.3%

救済率 ··· 正しい文字に訂正した数／認識誤りの数

誤訂正 ··· 正しく文字認識できていた文字を誤った文字に訂正してしまった場合

3.2 考察

郵便番号を利用したことの効果 地名部の照合に郵便番号を利用したことでは、十分な効果が得られた。この処理によって、地名部分を正しく訂正できた数は260件 (全体の37.8%) であった。また処理速度については、対象となる文字列が長いほど郵便番号を用いた効果が顕著である。特に都道府県のレベルから書き始められている住所の場合は、郵便番号による処理を併用すると、平均処理時間を12%に短縮できた。

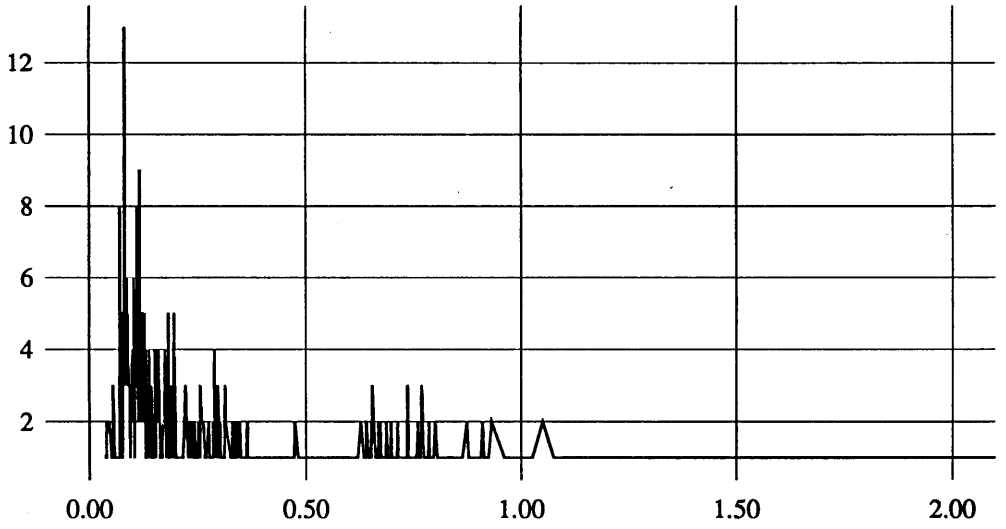


図 6: 処理時間の分布

住所構造解析の処理時間　構造解析処理の処理時間の分布をみると、図6のように、密になっているところが2ヶ所ある。一つは住所構造の候補が1つだけしか生成されなかった場合で、この部分だけで全体の63%である。次が2つの住所構造が生成された場合である。これらを合わせて全体の90%になる。これ以上の処理時間がかかっているデータは、特殊な場合である。たとえば最も処理時間がかかったものは、「栄区飯島町1－2」と書かれた帳票の認識結果データであったが、この場合「栄」の認識結果の候補文字中に正解文字がなく、そのため日本全国の2文字の区それぞれについて町名の検索を行なってしまったからである。一般に、文字列長が短く、文字認識の精度が低いデータほど処理時間が余計にかかる傾向がある。

切り出しキー文字が見つからない場合　住所の区切り文字をキーとして処理を行なう本方式では、認識結果文字列の中からキーとなる文字が欠落していた場合は、本方式の利点が生かせない。今回の入力データの中で、キー文字が候補文字から欠落しているものは20件、全住所中の3%であった。これらに対しては、住所構造知識に定義された単語長を用いて切り出し位置を、決めていくことができる。

地名部・番地部の境界が正しく判断できない　地名部の解析結果が複数あり、これらの長さが異なっている時、番地部の開始位置の決定方法が問題になる。地名部と番地部との境界を間違えてしまうと、正解を得ることはできない。689件の住所文字列データを処理した結果、31件(4.5%)が地名部の終端位置の決定に失敗している。

番地部の開始位置を確実に合わせるためには、地名部の解析結果に長さの異なるものがある場合、その各々について番地部の処理を行なう必要がある。また別の方法として、番地部の処理を文字列の終端から逆方向に行ない、地名部との境界を最後に調整することが考えられる。

方書フィールドの処理精度が極端に悪い　地名部に表れる単語は、その大部分が地名辞書に格納されている。これに対し、今回の評価で用意した方書用辞書は非常に小規模であったため、帳票に書かれた住所の方書は大部分が辞書未登録語となってしまった。会社・学校・ビル・マンション・アパート・個人名などを集めた方書専用の単語辞書を用意することで、番地部文字列単位で数十%の処理精度の向上が見込める。しかし、地名部のようにその構造を利用して照合範囲を決定することができないため、更に処理精度をあげるためには、文字種の変化点を単語の切れ目として照合範囲を決めるなどの工夫が必要となる。

数字の訂正に失敗することが多い　番地部の処理では、数字を読んでいるのに認識結果の中に数字候補が含まれていないことがある。特に評価実験で使用したOCRの場合は文字種を制限しないで、認識を行なった場合、英数字や記

号のような少画数文字の認識率が低いという傾向がある。現在の機能では番地や部屋番号の数字に対して、候補文字に正解が含まれていないと誤読文字を訂正することが出来ない。

評価結果によると、候補文字に正解が含まれていないという現象は、特に「1」の場合に多く現れた。これに対処するために、数字であるべき文字位置に「／」「ノ」「メ」などが候補に表れた場合は、強制的に数字の「1」にしてしまうような処理(参考文献[1])や、認識辞書を数字のみに制限して認識を行なった結果を用いる処理が必要となる。

地名辞書に登録されていない地名がある　帳票を正しく読み取っていても、照合処理に失敗することがある。これは町名変更などに合わせて、地名辞書の内容を更新していないため、正しい単語が検索できないからである。このような場合が、今回の入力データでは全942件のうち2件確認できた。1989年版郵便番号簿によると、1987年から1989年の間に47都道府県すべてで市町村名と郵便番号に対して、何らかの変更が行なわれている。このため実際の運用時には、地名辞書を町名変更に合わせてアップデートしていかなければならない。

4　まとめ

本稿ではの帳票の住所文字列を文字認識した結果に対して、自動的に誤りを検出・訂正するシステムについて報告した。試作システムでは、現在の処理方式のままでも閾値等のパラメータの最適化によって、処理精度の向上は期待できる。地名部については実用的な数値が得られたが、システム全体の大幅な性能アップのためには、考察で述べたように番地部分の処理精度を向上させなければならない。特に未知語が数多く現れる方書フィールドの照合方法が課題となっている。

参考文献

[1] 清野和司, 柳楽さつき, 中尾和則 「自由記載住所文字列に対する知識処理」 平成元年電子情報通信学会春季全国大会 D-465

[2] 鈴木章, 宮原末治, 小橋史彦 「手書き住所認識の後処理法」情報処理学会第38回全国大会 1K-1

[3] 磯山秀幸, 木谷強 「OCRの認識結果に対する文字認識後処理方式の検討」 情報処理学会第40回全国大会 2E-3

[4] 磯山秀幸, 木谷強 「住所の文字認識結果に対する後処理方式の検討」 情報処理学会第41回全国大会