

2N-11

要素単語の相互チェックに基づく

手書き文字列認識知識処理

下村秀樹

NEC情報メディア研究所

1. はじめに

枠なし手書き文字列の読み取りでは、単独文字の切り出しや認識が難しい。そのため、実用化に際しては対象を限定し、強い知識からの制約で認識候補を選択・補正することが行われている。代表例として、住所、帳票の読み取りなどが挙げられる[1][2]。

記入枠制限の有無にかかわらず、従来行われている知識処理は、(1)辞書との単語照合、(2)単語間の接続関係チェック、の2点を中心であるが、特定の読み取り対象について、複数単語間の冗長性に着目した知識処理が考えられる。例えば、文献[2]では枠あり帳票の読み取りにおいて、氏名漢字に対してふりがなという冗長な情報を照合し、飛躍的に読み取り性能を上げたと報告されている。正解を格納したテーブルと認識候補文字とを直接比較して、単語を相互に補正し合っている。

ただし、上記の処理を枠なし手書きに直接応用することは難しい。文献[2]のように記入枠ありでかつ記入単語の種類(例えば、漢字氏名、ふりがななど)が指定されているという条件下では、文字切り出しの曖昧さがなく、各認識候補文字と正解レコード中の文字との対応も決まっている。したがって、正解の要素単語の組合せを登録したテーブル(正解テーブル)と文字単位で直接照合することが比較的容易である(図1)。これに対し枠なし手書きの場合

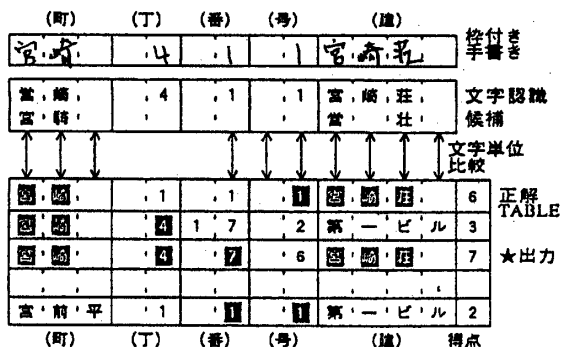


図1 単語間の冗長性に基づく知識処理(記入枠あり)

A knowledge processing based on mutual checking among words for handwritten string reading

Hideki Shimomura

NEC Corporation

は、文字の切り出しが複数候補になるとともに、どの種類の単語がどこに書かれているのかが事前にはわからない。文字の切り出しと認識多候補をふまえて、さらに単語の種類の可能性を考慮し、図1のように文字を基本としてテーブルとの照合を行うと、処理量が非常に多くなるという困難さがある。

しかし、文字切り出しや文字認識の難易度は枠なし手書きの方が圧倒的に高く、単語間の冗長性を利用した知識処理の効果は大きいと考えられる。そこで、枠なし手書きでも要素単語の相互チェックに基づく知識処理を可能とするために、文字認識候補から要素単語候補をいったん抽出して要素単語の種類を決めた後、単語レベルで正解テーブルと照合する処理方式を提案する。

2. 基本処理ステップ

提案する処理の流れは次の通りである。知識処理に先立ち、文字領域の切り出しと認識を行って、候補文字が出されているとする。

ステップ1: 切り出された候補文字群と単語辞書との照合を行い、単語を抽出する。候補文字群に正解が含まれている保証はないので、一部が欠けていても候補として挙げ、尤度付きで出力する。手書き文字列に対して、DPマッチングをベースとした有効な方式がすでに提案されている[3]。

ステップ2: 候補単語をキーとして正解テーブルを検索し、一致したレコードの番号を得る。

ステップ3: 検索されたレコード番号に対応する得点カウンタに、検索キーに使用した候補単語の尤度を加算する。要素の重要度に従って加算の重みを変えてもよい。

ステップ4: 加算された尤度の上位候補を取り出し、競争候補の判定を行って最終結果を出力する。

3. 読み取りへの適用例

図2では、住所の町名・番地部分に対する建物名の冗長性を利用した処理の例を示している。

まず住所・建物名单語辞書と番地記載に関連する知識に基づき町名、番地、建物名の単語を抽出する。この例では、町名と建物名は独立に単語辞書を検索しているが、番地部分は町名の直後から生成規則を参照しながら候補を作る。

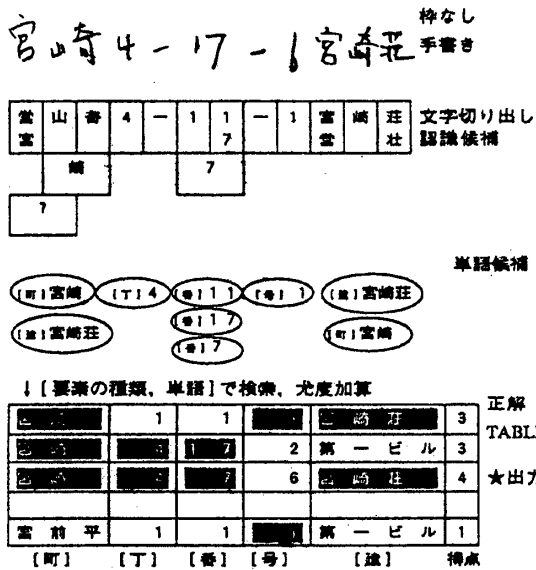


図2 単語間の冗長性に基づく知識処理 (記入枠なし)

さらに番地部分は、丁目などの各階層を一つの単語として扱う。

次のステップでは、各単語をキーとして正解テーブルを検索し一致したレコードにキーの認識尤度を加算してゆく。例では「宮崎」という地名、「宮崎荘」という建物名がそれぞれ2箇所から抽出されているが、二重に加算する必要はなく、値の大きい方について1回だけ行えばよい。なおこの例では、どの単語も尤度を1として計算している。

最後に、得点が上位のレコードを取り出し、競合の判定を行う。必要があれば、要素間の記載位置整合性のチェックも行い、適切と思われる候補を出力する。この例では、複数の文字切り出し/認識の候補の中から、得点が4点の「宮崎4丁目7番地6号宮崎荘」が正しく選ばれた。文字認識では番地部の3階層目が誤読されているが、単語の冗長性により補正されていることがわかる。

#### 4 議論

##### 4.1 単語階層を抽出する利点

提案方式では、要素単語の組合せで表現されたテーブルのレコードと認識候補文字を直接比較するのではなく、いったん単語を抽出している。そもそも枠なしの手書き文字列の場合、要素の種類を決めなければ正解テーブルとの照合量が爆発的に増大するので、それを避けるために単語抽出は必須の処理とも考えられる。しかし、いったん単語を抽出することで、異なるレコード中の同一単語と認識候補の照合処理を繰り返し行う必要がなくなるという利点も

ある。例えば、図2の「宮崎」という町名は正解テーブルの複数のレコードに現れているが、認識候補文字との照合は単語抽出時の1回だけで、あとは単語をキーとして正解テーブルを検索することになる。検索では、一般のデータ検索などで用いられる技法を流用することが可能である。

##### 4.2 得点加算/上位候補抽出処理の効率化

この手法で問題になるのは、主に得点の加算、および上位候補抽出の処理時間である。まず得点の加算では、候補単語をキーに検索されたレコードに尤度を加算してゆくが、非常に多くのレコードに一致する単語の場合、レコードの限定能力がないにもかかわらず処理時間は多くかかり無駄が多い。これに対しては、レコード限定能力の高い単語によって、まずテーブル内のレコードを限定してしまうといった方策が考えられる。また、上位候補の抽出に際しては、予め閾値を設定しておき、尤度加算時に閾値を超えたレコードの番号を別の領域に記憶しておくことで上位候補抽出対象を限定するような手法も有効である。

##### 4.3 頑健性

またこの手法は、文字切り出しおよび認識候補とレコードのすべての組合せを照合しているわけではない。単語抽出の段階で出力できる候補数は限られている（だからこそテーブルと照合が行える）ことから、ここで正解単語が切り捨てられる場合があり、最良解は保証されない。しかし、複数の要素単語の冗長性によって、切り捨てられた単語が補完されることもあり、一つの単語の欠落が致命的にはならないという頑健性がある。したがって、記載時の省略にも対処可能である。ただし、補完を無条件で信じることは、偶発的な誤読を産む原因ともなる。補完で推測した単語については、再度文字認識を行って確認するなどの枠組みが必要であろう。

##### 5. おわりに

本稿では、要素単語の冗長性を利用した手書き文字列読み取り知識処理方式について述べた。今後、実問題への適用をめざし、基礎的検討を進める。

##### 参考文献

- [1]清野他:自由記載住所文字列に対する知識処理,1989信学全大D-465
- [2]浅見他:ふりがなを利用して認識率を上げた手書き漢字OCR,日経エレクトロニクス,1988.10.17
- [3]福島他:手書き文字列読み取りのための単語列検索アルゴリズム-文字タグ法-,情処論文,Vol37, No.4,1996