3U-8

制約充足による手書き変体仮名認識の構想

新井 侑太† 鈴木 徹也‡ 相場 亮‡
†芝浦工業大学大学院電気電子情報工学専攻 ‡芝浦工業大学システム理工学部
{ma11011, tetsuya, aiba}@shibaura-it.ac.jp

1 はじめに

現在では, 国文学における研究でも計算機上で検索・閲覧できる資料を活用するのは当然となっている [2]. しかし, 資料を翻刻しテキストデータ化するには多くの時間と労力が必要となる. 例えば古事類苑の全文入力 [3] では画像データを見ながら手入力で翻刻をしている. また, 資料の中には変体仮名 (図 1) や漢文で書かれているものが多い. 特に仮名には「くずし字」と「踊り字」という特徴がある. くずし字は形が崩れているために, 楷書を対象とした文字認識は利用できない. 踊り字は直前の一文字, または二文字と同じ読みをする文字である. 踊り字のみで読みが決定しないため一文字単位の認識手法は利用できない. これらのことから, 従来の文字認識を用いることは難しい.

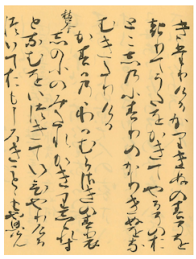

図 1: 本研究で扱う変体仮名の例 [1]

2 先行研究

手書き仮名文字認識の先行研究 [4, 5] では, 予め用意した文字画像の特徴量を利用する. 認識結果を文字単位で決定する方法や n-gram を用いて連続する文字を考慮する方法等がある. 文献を特定のものにすることにより, およそ 80〜90%の認識率を実現している. 認識には画像から抽出した特徴量を用いている (図 2).

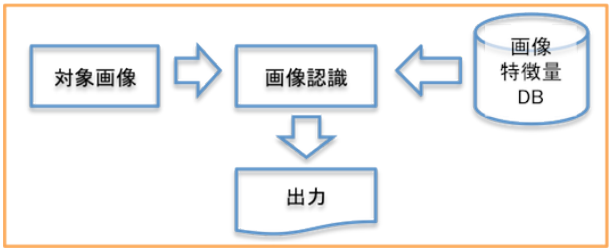

図 2: 従来の認識手法の概要

先行研究は特定の文献に特化しているため汎用性に欠ける. 別の文献に対して使用する場合は画像特徴量を学習する必要がある. また, 踊り字の認識は行われていない.

3 制約充足による手書き変体仮名認識

先行研究よりも汎用性を持った文字認識を実現するために我々が構想している文字認識の枠組み (図 3) を説明する. 先行研究との相違点は, 制約解消器を用いて, 画像認識器の結果を補正する点にある. 画像認識器が制約充足問題を構築し, 制約解消器がその解を求める. 制約解消器がその解を画像認識器へフィードバックすることで, 画像による文字認識を補正する. これを必要なだけ繰り返し, 最終的な解を出力する. 局所的な文字の並びに関する条件には単語辞書を用いる. 単語辞書には対象とする資料の年代に合わせた辞書を利用できる. 離れた文字の類似性と踊り字に関する条件には制約を用いる.

この構想の技術的課題は画像認識に関する課題と制約解消に関する課題とに大別できる. 画像認識に関する課題には, 画像からの文字の切り出し法と, 切り出した文字の認識方法とがある. 制約解消に関する課題には, 翻刻作業の制約充足問題としてのモデリング方法と, 効率的な制約解消法とがある.

現在我々は主に制約解消に関する課題に取り組んでいる.

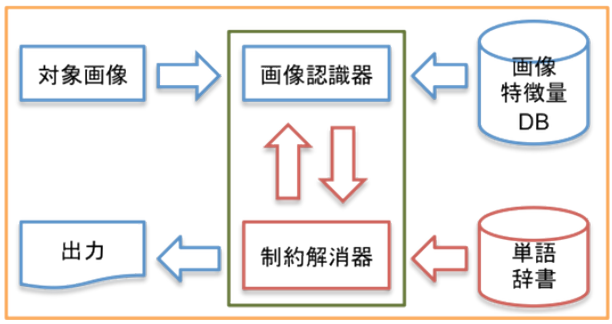

図 3: 提案構想の概要

4 制約充足問題

制約充足問題の構成要素は, 変数とその領域, 変数の値を制限する制約である. 提案構想における制約充足問題の概要は以下の通りである.

変数 画像認識器が切り出した一文字分の画像領域に対応する. その変数の領域は, 画像認識器が決定したその画像領域の候補文字である. ある画像領域を一文字として翻刻するか, 二文字として翻刻するか判断できない場合がある. その場合には, 両方の可能性を考慮して変数とその領域を設ける. 画像認識器は, 最後の文字に対応する変数を除いて, 各変数に後続の変数を与える. これにより変数上に半順序が定義される.

局所的な文字の並びに関する制約 半順序で定義される変数列と単語辞書内の単語とを用いる. 離れた文字の類似性の制約, 踊り字の制約, 各変数が不可読文字でないという制約には, 等号と非等号を用いる. 不可読文字とは, 画像認識器の候補文字が誤っていることを示す特別な文字である.

制約の優先度 制約階層 [6] を導入し, 制約に充足優先度を与える. 踊り字の制約に最も高い優先度を与える. 各変数が不可読文字でないという制約には最も低い優先度を与える. 制約階層の比較器で極大となる解を画像認識器へのフィードバック及び最終出力とする.

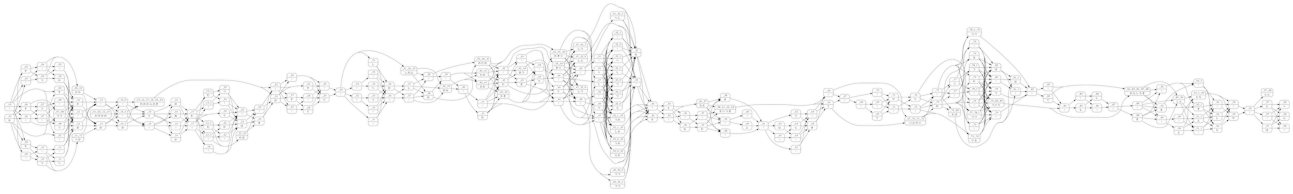

図 4: 読みの割当結果のグラフ

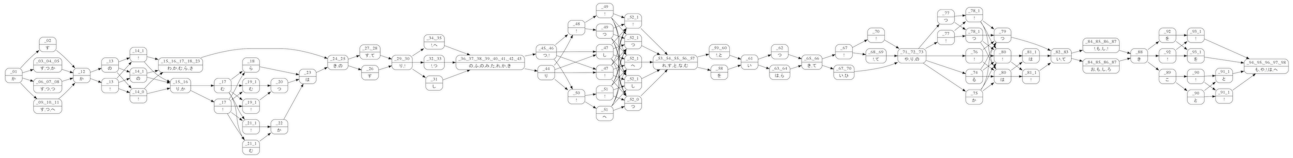

図 5: 探索空間の削減結果のグラフ

5 制約解消器

我々は 4 節で説明した制約充足問題の解を求める制約解消器を実装した. 使用した言語は Ruby である. その解消器は次のように動作する.

1. 単語辞書を参照して, 可能な読みの割当 (変数列とそれに合致する単語との組) を全て求める.
2. 読みの割当を操作し, 探索空間を削減する.
3. 分岐限定法を用いて解を探索する.

図 4 は読みの割当結果を表すグラフの例である. 一つの頂点で変数列とそれへの読みの割当を表している. 図 5 は図 4 を基に探索空間を削減した例である. 図 5 の左端の頂点から右端の頂点へ至るパスを一つ求めると, そのパス上の変数の値が一意に決まる. 手順 3 ではそのようなパスのうち, 制約充足度が極大となるパスを求める.

実装した制約解消器を用いて予備実験を行った. 目的は探索時間がどの程度かかるかの確認である. 対象は文献 [1] の一部とした (図 1). 詳細な位置は, 3 頁の 1 行目「かりきぬの . . .」から 8 行目末までの 100 文字である. 認識候補は人手で作成した. 単語辞書に登録された単語数は 241 語である. 使用した計算機の CPU は Intel Core i7, クロック数 2.7GHz, 搭載メモリ 4GB である. 結果を表 1 に示す. 表 1 の値は 10 回試行を行い, 合計時間が最大の結果と最小の結果を除いた平均である.

表 1: 探索時間の計測結果

読みの割当の個数	1536
辞書読込と読みの割当 (秒)	0.0002
探索空間の削減 (秒)	0.0989
解の探索 (秒)	503.5448
合計時間 (秒)	504.2289

表 1 から計算時間全体の約 99%が探索であることがわかる. また, 汎用性を考慮すると単語辞書の登録単語数増加が予想されるため, 読みの割当の個数も増加し, 計算時間は長くなると考えられる.

6 まとめ

国文学における資料を画像データからテキストデータへ翻刻する上で重要となる文字認識の新しい構想について述べた. 現在は制約解消に関する課題に取り組んでいる. 制約充足問題を定義し, 離れた文字の類似性の制約, 踊り字の制約, 各変数が不可読文字でないという制約を設けた. その問題を解くために制約解消器を Ruby 言語で実装した. 予備実験として探索時間を調べた. 100 文字で約 8 分の処理時間がかかった. 今後はグラフの分割などによる探索の高速化, 画像認識器の実装, 画像認識器と制約充足器の連携を行っていきたい.

参考文献

[1] 鈴木知太郎. 御所本伊勢物語 冷泉為和筆 宮内庁書陵部蔵 影印本. 笠間書院, 1994-4-30.

[2] 渡上将治, 村川猛彦, 宇都宮啓吾, 中川優. 文献調査支援のためのスタンドアロン型全文検索システムの構築. じんもんこん 2011 論文集, pp. 225–230, Dec 2011.

[3] 山田奨治, 早川聞多, 相田満. 古事類苑 (天部・地部) の全文入力と Wiki 版の試行 : 前近代の文化概念の情報資源化. 情報処理学会研究報告. 人文科学とコンピュータ研究会報告, Vol. 2006, No. 112, pp. 39–46, 2006-10-27.

[4] 山田奨治, 柴山守. n - gram と OCR による定型表現がある古文書の文字の推定. Technical Report 59(2003-CH-058), 国際日本文化研究センター・研究部, 大阪市立大学・学術情報総合センター, May 2003.

[5] 和泉勇治, 加藤寧, 根元義章, 山田奨治, 柴山守, 川口洋. ニューラルネットワークを用いた古文書個別文字認識に関する一検討. 情報処理学会研究報告. 人文科学とコンピュータ研究会報告, Vol. 2000, No. 8, pp. 9–15, 2000-01-21.

[6] Alan Borning, Bjorn Feldman-Benson, and Molly Wilson. Constraint Hierarchies. *Lisp and Symbolic Computation*, Vol. 5, No. 3, pp. 223–270, 1992.