

手書き変体仮名認識における制約充足問題の拡張

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

近年、国文学研究においてもコンピュータを活用することが当然となっている [2]. しかし、資料を翻刻して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や漢字で書かれているものが多い (図 1)。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手法は利用できない。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、従来の文字認識手法を用いることは難しい。そこで我々は制約充足を用いた文字認識手法を提案した [4, 5]。現在は、特に制約解消に注力して研究を行っている。本論文では提案した文字認識手法で用いる制約充足問題に新たな制約を導入し、その効果を確認する。

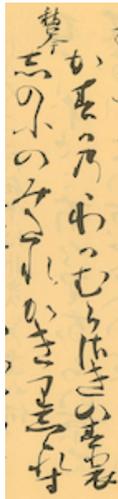


図 1: 伊勢物語の一部 [1]

2 制約充足による文字認識手法

文献 [4, 5] で提案した文字認識手法の概要 (図 2) を説明する。この文字認識手法は大きく分けて画像認識器、制約解消器、単語辞書、画像特徴量データベースから構成されている。

簡単に認識手順を説明する。まず、画像認識器は入力画像から読みの候補を求め、読みの候補から制約充足問題を構築する。画像認識器は構築した制約充足問題を制約解消器に渡し、制約解消器がその解を求める。制約解消器が解消結果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。

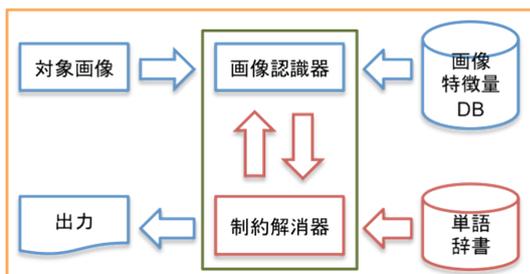


図 2: 制約充足による文字認識手法の概要

3 翻刻制約充足問題

文献 [4, 5] では翻刻支援のための制約充足問題として翻刻制約充足問題を定義した。翻刻制約充足問題は有向非循環グラフで表すことができる。そのグラフのノードは図 3 の様に変数とその領域を表す。グラフのエッジは文字の読み順を表す (図 4)。



図 3: 変数とその領域を表すノード

3.1 変数とその領域

変数は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読みの候補である。領域は、ひらがな全体に特殊記号 “!” と “?” を追加した集合の部分集合である。“!” は画像認識器が決定した候補が誤っていることを表し、不可読文字と呼ぶ。“?” は平仮名ではない文字を表す。

3.2 制約

制約は暗黙的制約と明示的制約の 2 通りに分けられる。暗黙的制約は明示的に記述しなくてもよい制約である。読み順制約、単語列制約の 2 つがある。読み順制約は文を読む際にたどる変数の順番を表す。単語列制約は、読み順制約を満たす一連の読みが、単語列の部分列であることを表す。

明示的制約は明示的に記述する制約である。等号制約と等号否定制約の 2 つがある。等号制約は読みが等しいことを表す。字形が類似している文字同士や踊り字に用いる。等号否定制約は読みが異なることを表す。

3.3 制約階層

翻刻制約充足問題では制約同士が競合し、全ての制約を同時に満たせない状況 (制約過多) が生じうる。制約過多に対応するため制約階層 [6] を導入し、各制約に優先度を与えた。これにより優先度の高い制約を優先的に充足できる。解の比較には locally-predicate-better [6] を用いて最適解を求める。

4 制約解消法

翻刻制約充足問題の制約解消手順を以下に示す。

1. 制約充足問題グラフ (例: 図 4) と単語辞書を参照して、可能な読みの割り当て (変数列とそれに合致する単語との組) を全て求め、読みの割り当てグラフを作る。そのグラフのノードは変数列とその変数列に割り当てる文字列の組を表す。
2. 読みの割り当てを操作し、読みの割り当てグラフを短縮する。
3. 分岐限定法を用いて解を探索する。

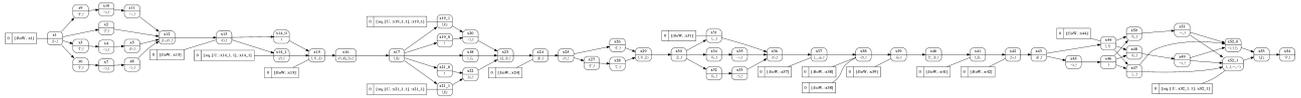


図 4: 翻刻制約充足問題のグラフ表現

5 単語開始制約と単語終了制約の導入

3 節の制約のみを用いた文献 [4, 5] の実験結果は、最適解が多く、探索に時間がかかっている。そこで新たな制約を導入することで解の絞り込みを目指す。導入する制約は単語開始制約と単語終了制約である。

単語開始制約と単語終了制約は、それぞれ単語の開始文字と終了文字を指定する制約である。新しい制約を扱えるようにするために変数の付値 (文字) に属性を持たせる。その属性は文字が単語内でどの位置にあるかを表す。属性値には開始、中間、終了、不明の 4 種類がある。

6 制約解消実験

新制約の効果を確認するために従来の翻刻制約問題 (csp) と拡張した翻刻制約充足問題 ($csp_{n,m}$) とを解き、制約解消時間と最適解数を比較する。なお、 n は単語開始制約数、 m は単語終了制約数を表す。

実験では 4 種類の結果を比較する。1 つ目は csp 、2 つ目は $csp_{0,0}$ 、3 つ目は $csp_{3,3}$ 、4 つ目は $csp_{6,6}$ である。認識対象文として伊勢物語 [1] から図 1 の「かすがのわかむらさきのすり衣しのぶのみだれかぎりしられず」(31 文字) を使用した。画像認識器の出力である認識結果は人手で作成し、認識結果から翻刻制約充足問題を作成した。4 節で述べた手順 2 が新制約に対応できていないため、全ての翻刻制約充足問題で手順 2 は省略した。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。制約解消器は Ruby 言語で実装した。制約充足問題のサイズを表 1 に、最適解の個数と各種処理時間を表 2 に示す。

表 1: 翻刻制約充足問題のサイズ

	csp	$csp_{0,0}$	$csp_{3,3}$	$csp_{6,6}$
制約充足問題	58	58	58	58
グラフのノード数	58	58	58	58
読みの割り当て	136	138	138	138
グラフのノード数	136	138	138	138
制約数	62	62	68	74

表 1 の読みの割り当てグラフのノード数を見ると、 $csp_{n,m}$ の方がノード数が 2 個増加している。これは同一変数に同じ文字を割り当てるノードが、その文字の属性値によって区別されるためである。

表 2 の最適解の個数を見ると、 $csp_{0,0}$ は csp の約 4 倍に増加している。 $csp_{3,3}$ は csp の約 2 倍になっている。 $csp_{6,6}$ は csp の約 0.88 倍に減少している。探索時間を見ると、 $csp_{0,0}$ は csp の約 4 倍に、 $csp_{3,3}$ は約 2 倍に延

表 2: 最適解の個数と各種処理時間

	csp	$csp_{0,0}$	$csp_{3,3}$	$csp_{6,6}$
最適解の個数	64,896	249,600	129,792	43,264
辞書読込と読みの割り当て (秒)	0.47	0.49	0.46	0.43
解の探索 (秒)	632.20	2679.79	1452.26	556.75
合計時間 (秒)	632.67	2680.28	1452.72	557.18

びいるが、 $csp_{6,6}$ は csp の約 0.67 倍に短縮されている。この結果から新制約は一定数以上使用することにより効果が発揮されると言える。

7 まとめ

本論文では翻刻制約充足問題の拡張について述べた。新制約として単語開始・終了制約を導入した。新制約の導入に伴い、変数の付値に単語内位置を属性として与えた。従来の翻刻制約充足問題と拡張した翻刻制約充足問題で制約解消実験を行い、単語開始・終了制約の効果を調べた。単語開始・終了制約は一定数以上用いることで効果が期待できることがわかった。しかし、31 文字分の読みの候補として 4 万通り以上が出力されているため、さらに最適解の絞り込みが必要である。今後の課題としては探索空間の削減と単語開始・終了制約の個数による効果変動調査とが挙げられる。

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] 新井侑太, 鈴木徹也, 相場亮. 制約充足による手書き変体仮名認識の構想. 第 74 回全国大会, 2012-3-7.
- [5] Yuta ARAI, Tetsuya SUZUKI, and Akira AIBA. Recognizing historical KANA texts using constraints. *Proceedings of The Second Workshop on Computation: Theory and Practice*, 2012-9-27.
- [6] Alan Borning, Bjorn Feldman-Benson, and Molly Wilson. Constraint Hierarchies. *Lisp and Symbolic Computation*, Vol. 5, No. 3, pp. 223-270, 1992.