1Q-2

手書き変体仮名認識における単語の連接関係を利用した解の絞り込み

渡辺 悟 † 鈴木 徹也 ‡

† 芝浦工業大学 システム理工学部 電子情報システム学科 ‡ 芝浦工業大学 システム理工学部 電子情報システム学科

1 研究の背景と目的

古文書で使われている仮名は変体仮名と呼ばれ現代の仮名とは異なる. 変体仮名は同じ読みをする文字が複数存在していたり, 同じ文字でも写本によって形が異なっていたりする. そのため, 古文の翻刻作業には多くの知識と労力を必要とする. そこで, 我々の研究グループでは翻刻支援システムの研究をしている.

我々の研究グループの新井は制約充足による手書き変体仮名認識 [1] を提案した. 手書き変体仮名認識において, 画像特徴量による文字認識だけでは正しい認識結果を得るのが困難である. そこで新井は画像特徴量による文字認識では読みの候補を挙げることにし, 読みの組み合わせの中から最適な単語列を求めることにした.

新井の方法では, 制約 (変数の値の条件) が少ないと組み合わせ爆発を起こし, 最適解の数と計算時間が膨大になることがあった. 制約を自動で加えることは難しく, また現在は限定的な単語辞書を用いているが, 汎用的な単語辞書を用いれば解はさらに増える. 人間の支援システムである以上, 解の数はなるべく少ない方が望ましい. よって, 別の方法で解を絞り込む必要がある.

本研究では, 正しい認識結果に近い解が残るように最適解の数を削減することを目的とする.

2 制約充足による手書き変体仮名認識

新井による先行研究を簡単に紹介する.

2.1 制約充足問題

制約充足問題は変数とその領域, 制約から構成される. **制約**とは, 変数が同時にとることのできる値の集合を表す条件のことである. 変数への値の割り当てを**解**と呼ぶ.

2.2 翻刻制約充足問題

新井は翻刻制約充足問題という翻刻支援を目的とした制約充足問題を定義した. 変数は入力画像中の変体仮名, その領域は読みの候補となる. 制約には以下のものがある.

変数と読みの並びの制約 読みは単語の列になる
等号制約 2 つの変数または定数の値が等しい
等号否定制約 2 つの変数または定数の値が異なる

Narrowing Down of Solutions Using Adjacency Relation of Words in Recognizing Historical KANA Texts
Satoru WATANABE†, Tetsuya SUZUKI‡
†Department of Electronic Information Systems
College of Systems Engineering and Science
Shibaura Institute of Technology
‡Department of Electronic Information Systems
College of Systems Engineering and Science
Shibaura Institute of Technology
{p10109, tetsuya}@sic.shibaura-it.ac.jp

単語開始制約 ある変数は単語の開始位置にある
単語終了制約 ある変数は単語の終了位置にある

制約には優先度があり, より優先度の高い制約を満たす解が最適解となる.

2.3 読みの割り当てグラフ

読みは単語の列になるという制約を満たすために, 単語辞書を参照して可能な読みを割り当てたグラフ (**読みの割り当てグラフ**) が作成される. 読みの割り当てグラフは有向非循環グラフで表され, ノードが変数への読みの割り当てに対応し, 有向リンクが読み順を表す. 最上流ノードから最下流ノードへのパスに沿って得られる変数への読みの割り当てが解の一つになる. 図 1 は読みの割り当てグラフの一部である. 各ノードは 1 つの単語を表しており, 上部に変数名, 下部にその変数へ割り当てられた文字が表記されている.

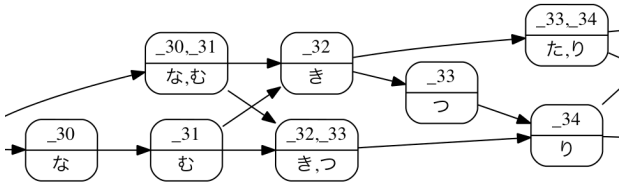

図 1: 読みの割り当てグラフの一部

2.4 解の探索方法

従来手法の解の探索方法を以下に示す.

1. 翻刻制約充足問題から読みの割り当てグラフを作成する.
2. グラフを簡約する.
3. 簡約したグラフの最上流ノードから最下流ノードに向かってノードを訪問し, 分枝限定法を利用して, 最適解を探索する.

制約が少ないとき, 十分な枝刈りが行われず, 組み合わせ爆発が起こる.

3 単語の連接関係を利用した解の削減

コスト最小法 [2] と N-best 探索 [3] を用いて解を絞り込むことを提案する.

3.1 コスト最小法

コスト最小法は, 単語の生起コストと連接コストを定義し, コストの総和が最小となるパスを最適解として選択する方法である. 生起コストはノードに与えられ, その単語が表れにくいほど高くなるようにする. 連接コストはリンクに与えられ, 両端のノードの単語が連接しにくいほど高くなるようにする.

3.2 N-best 探索

最短経路問題で上位 N 個の最適解を求めることを N-best 探索という. 前向き DP(動的計画法) 後向き A*

アルゴリズムによって効率良く求めることができる．アルゴリズムは後述する．

3.3 解の探索方法

提案手法の手順を以下に示す．

1. 翻刻制約充足問題から読みの割り当てグラフを作成する．
2. 読みの割り当てグラフを読み込み, 辞書を引いてグラフを再構築する．
3. コストの総和が少ない順に上位 N 個の解を求める．
4. 解の中からより優先度の高い制約を満たす解を残す．

手順 2 でグラフを再構築しているが, これには 3 つの理由がある．従来手法では単語の種類 (品詞など) を区別していないので, それを区別するためと, 文頭であることを表す文頭ノード (以下文頭) と, 文末であることを表す文末ノード (以下文末) を追加するためと, ノードとリンクにコストを付与するためである．

手順 3 について説明する．まず, 手順 2 で構築したグラフの文頭から各ノードまでの最小コストを動的計画法で求める．最小コストは, トポロジカル順序で上流から下流に向かってノードを取り出し, 各下流ノードの対して最小コストを更新していくことで求まる．

最小コストが求まったら, 文末から A*アルゴリズムを応用して探索を行い, 上位 N 個の解を求める．ここで 4 種類のコストの記法を定義する．文末からあるノード n を通って文頭に到達するときの最小コストを $f(n)$, 文末からノード n までの最小コストを $g(n)$, ノード n から文頭までのコストを $h(n)$, ノード n の生起コストを $c(n)$ とする．するとその 4 種類のコストの間に $f(n) = g(n) + h(n) - c(n)$ が成り立つ．$h(n)$ は前段階で求まっているので, $f(n)$ が最小のノードから探索していけば, 最小コストのパスが直ちに求まる．そのアルゴリズムを以下に示す．ただし以下の手順中の $cost(n, m)$ はノード n とノード m の連接コストである．また, 探索木のノードを探索ノードと呼び, 探索ノード x が参照している読みの割り当てグラフのノードを $n(x)$ とする．

1. 探索ノードのリスト q を空にする．
2. 探索ノード s を q に加える．
   ($n(s) :=$ 文末, $f(s) := h(n(s))$ とする．)
3. q が空, もしくは得られた解の個数が N なら探索を終了する．
4. q から最小の $f(x)$ を持つ要素 x を取り出す．
5. $n(x)$ が文頭の時, 親を辿って解を得て 3 に戻る．
6. $n(x)$ の各上流ノード m に対して以下の操作を行う．
   (a) 探索ノード y を用意する．
   (b) $n(y) := m$ とする．y の親を x にする．
   (c) $f(y) := g(x) + h(m) + cost(n(x), m)$ とする．
   ($g(x) = f(x) - h(n(x)) + c(n(x))$)
   (d) y を q に加える．
7. 3 に戻る．

4 実験

従来手法との性能比較や, 辞書の変更などにより結果がどう変化するのかを調べるために実験を行った．対象は伊勢物語 [5] の一部とし, 翻刻制約充足問題は人手で作成した．単語の生起コストと連接コストは中古和文 UniDic[4] のものを使用した．使用した計算機の CPU は Intel Core i5, クロック数 1.8GHz, 搭載メモリ 8GB である．

従来手法と提案手法による実験結果をそれぞれ表 1, 表 2 に示す．どちらも Ruby 言語で実装している．辞書は対象に含まれている単語と基本的な助詞, 助動詞からなる辞書 (241 語) を使用している．ise2_1, ise2_2, ise2_3 は ise2 を分割したものである．各文字の読みの候補は高々3 つである．

表 1, 表 2 から従来手法より解の数, 計算時間が少なく, より良い解が得られていることが分かる．ノード数が増加しているのは, 単語の種類を区別したためである．

表 1: 従来手法の実験結果

問題	ノード数	解の数	平均正解率	計算時間
ise2_1	152	41,472	0.922	257 秒
ise2_2	230	測定不能		
ise2_3	72	28	0.870	0.31 秒
ise2	449	測定不能		

表 2: 提案手法の実験結果 (N=10)

問題	ノード数	解の数	平均正解率	計算時間
ise2_1	268	10	1.000	0.39 秒
ise2_2	402	10	0.953	0.56 秒
ise2_3	144	10	0.932	0.20 秒
ise2	805	10	0.962	2.20 秒

表 3: 提案手法で中古和文 UniDic を使用した実験結果 (N=10)

問題	ノード数	解の数	平均正解率	計算時間
ise2_1	2,237	10	0.972	0.96 秒
ise2_2	4,053	10	0.939	1.47 秒
ise2_3	1,262	10	0.961	0.69 秒
ise2	7,567	10	0.958	3.63 秒

辞書を中古和文 UniDic から作成した辞書 (257,879 語) に変更して行った実験結果を表 3 に示す．表 2, 表 3 から辞書の変更によりノード数, 計算時間が増加していることが分かる．また, 平均正解率はそれほど減少していないので, 辞書を変更しても問題はないと考えられる．

ise2_1 で各変数の読みの候補を 4 つにして実験したところ (N=10), 正解率は 0.617 となった．したがって, 実用には精度の高い画像認識器が必要だと考えられる．

5 おわりに

手書き変体仮名認識について, 単語の生起コストと連接コストを導入し, コスト最小法と N-best 探索を用いて解を絞り込むことを提案した．従来手法より解の数と計算時間が少なく, より良い解が得られたことを実験により確認した．今後の課題として, 単語辞書にない単語の考慮や, 画像特徴量による文字認識結果の順位の利用検討が挙げられる．

参考文献

[1] 新井侑太. 制約充足による手書き変体仮名認識, 芝浦工業大学修士論文, 2013

[2] 奥村学. 自然言語処理の基礎. コロナ社, 2010

[3] 永田昌明. 統計的言語モデルと N-best 探索を用いた日本語形態素解析法. 情報処理学会論文誌, Vol. 40, No. 9, pp. 3420-3431, 1999

[4] 中古和文 UniDic ver.1.3 http://www2.ninjal.ac.jp/lrc/index.php?UniDic%2F%C3%E6%B8%C5%CF%C2%CA%B8UniDic

[5] 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