1ZF-07

手書き変体仮名認識システム
--制約解消器の Web サービス化--

山藤一輝† 鈴木徹也† 相場亮†
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 はじめに

古文書の翻刻は国文学の研究にとって基礎的な作業である.しかしそれには多くの知識と労力を必要とする.その理由の一つは, 古文書に使われている変体仮名にある.そこで我々の研究グループは翻刻作業の支援を目的として,制約充足による手書き変体仮名認識システムの構想を提案し[1],制約解消器の改善を行った[2].これらの研究成果を以降手書き変体仮名認識システムと呼ぶ.現在手書き変体仮名認識システムは制約解消器の部分が完成している.

本研究では手書き変体仮名認識システム用制約解消器の Web サービス化を行う. 今後開発予定のシステムの構成要素と制約解消器を疎結合しすること,インターネットを通じて制約解消器の機能を公開できるようにすることを目的とする,

2 関連研究

凸版印刷が翻刻に関する OCR 技術の論文を発表した[3].この研究では,切り出した文字の一つ一つに対して字形データベースと照合し,類似度が大きいものを読みとする.原理検証実験ではこの段階で精度は約 80%であった.

我々の研究グループが開発するシステムは,字形データベースによる翻刻では原理的に不可能な,文脈的にしか決定し得ない文字についての処理が可能になると期待できる.

3 手書き変体仮名認識システム

我々の研究グループは複数文字の並び方に関する情報を利用できれば汎用性の高い文字認識が行えると考えた.画像特徴量の段階で似ている文字同士を関連付けることで解の絞り込みが行える. これらを実現するためには制約を用いるのが有効であると考え,以下のようなシステムを提案した.

A System for Recognizing Historical Kana Texts: Development of a WebService for Constraint Satisfaction

†Kazuki SANDO, †Tetsuya SUZUKI, †Akira AIBA
†Department of Electronic Information Systems, College of Systems Engineering and Science, Shibaura Institute of Technology

3.1 システムの概要

図 1 はシステムの構成を図 1 に示す.

まず画像認識器が入力画像から特徴量を抽出し,用意されたデータベースを用いて各文字の読みの候補を列挙する.その結果から制約充足問題を作成して制約解消器に渡す.次に制約解消器が,単語辞書を参考にしながら制約充足問題を解く.

次に制約解消器が制約充足問題の解を画像認識器にフィードバックする.画像認識器はフィードバックを元に読みの候補を補正する.これらを必要なだけ繰り返し,最終的な解を出力する.

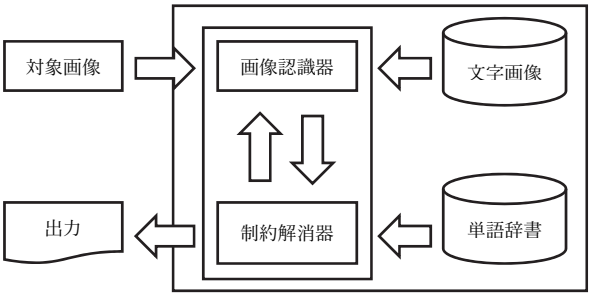

図 1:手書き変体仮名認識システムの構成

3.2 制約解消器

まず制約解消器は, 画像認識器で切り出した各領域を変数,読みの候補を領域とした制約充足問題を入力として受け取る.ここでの入力は変数名とその領域,グラフ作成に必要な変数間の連続関係,制約,使用する辞書名,n-best 探索の n の値である.次に辞書(中古和文 Unidic[4])を用いて可能な読みをすべて割り当てたグラフを作成する.この辞書はトライ木として実装するためダブル配列の形に変換される.その後,グラフに生起コストと連接コストを割り当てる.作成したグラフを利用して,コスト最小法で n-best 探索を行い,n 個の解候補の中で制約充足度が高いものを解とする.解は単語列になるように区切られた,変数とその領域の対である.この制約解消器はコマンドラインから利用可能となっている.

4 Web サービス化

制約解消期は Ruby で作成されている.よって本研究では Ruby on Rails を使用して Web サービスの開発を行った.

4.1 Web サービスの流れ

ユーザはサーバに制約充足問題,n-best 探索の

n,使用する辞書名の 3 つを http の POST メソッドで送る.サーバは受け取ったデータを制約解消器に渡す. 制約解消器は制約充足問題を解き,解をテンポラリファイルとして保存する.また,ランダムな文字列を含む URL をユーザに送る.その際にデータベースにそのランダムな文字列と解の組を登録する.

次にユーザ,先ほど受け取った URL に GET メソッドでアクセスする.サーバは受け取った文字列と紐づけられた解をデータベースから探索し,その解をユーザに送る.その後,先ほど利用したデータベースのレコードとテンポラリファイルを削除する.制約充足問題を解く際に正常に処理が終了できなかった場合は,解の代わりにエラーが出たことを示すメッセージを返す.また,データベースに登録された解が一定期間取り出されなかった場合は自動で削除する.

4.2 入出力

入出力で使用するファイルは JSON 形式である.POST で送るファイルの中身は制約充足問題,使用する単語辞書名, n-best 探索を行う際の n の値からなる.GET で受け取るファイルには解と満たされなかった制約が記述されている.ここで制約についての記述が必要となるのは,画像認識器で制約充足問題の補正を行う際に必要になると考えているためである.

5 実験

本実験では VirtualBox の仮想環境内にサーバを作成し,ゲスト OS とホスト OS との間で通信を行った.実験に使用した PC の CPU は Intel Core i7 2.3GHz,OS は Mac OS X El Capitan(10.11.6),メモリは 16GB 1600MHz DDR3 である.制約解消器単体で利用した場合と Web サービスとして利用した場合の所要時間の比較を行った.変数 199 個,制約が 208 個の制約充足問題を使用する.

表 1.制約解消器実行時間の比較

	コマンドラインからの使用	クライアント側の実行時間	Web サービス内の制約解消器実行時間
1 回目	3.73	137.4	134.6
2 回目	3.56	159.8	157.6
3 回目	3.80	138,0	135.0
4 回目	3.60	144.2	141.3
5 回目	3.57	141.6	137.0
平均	3.65	144.2	141.3

表 1 が実験結果である.制約解消器をコマンドラインから利用する場合は 3.65 秒で解を得ることができた,Web サービスとして利用した場合,制約解消器の所要時間は 141.3 秒であった.

Web サービス化した際に制約解消器の部分の所要時間が大幅に増加していたため,制約解消器の各工程の時間の測定を行った.表 2 がその結果である. 所要時間のほとんどが辞書を変換する部分と,グラフを作成する部分である.

表 2.制約解消器各工程の時間の測定

	制約解消器の実行時間	辞書ファイルの変換	グラフの作成
1 回目	142.2	79.9	43.1
2 回目	145.5	88.0	40.2
3 回目	144.6	96.3	31.6
4 回目	178.3	118.0	43.4
5 回目	154.6	93.3	41.2
平均	153.0	95.1	39.9

6 評価

Web サービスでの制約解消器は単体での利用に比べて非常に多くの時間がかかっており,所要時間のほとんどが制約解消器の部分である,特に時間がかかるのが辞書の変換とグラフの作成であるので,この部分を改善すればユーザの待ち時間の大幅な短縮が期待できる.

7 今後の課題

ユーザの待ち時間を減らすために,システムの改善が必要である.問題のある部分は,単語辞書を変換する部分と,グラフを作成する部分だとわかっているので,原因の調査を行っていきたい.また現在は制約解消器でエラーが起きた場合に,エラーが出たことを示すメッセージを返すが,具体的なエラーの内容や対処法までは記述されていない.現状で想定できるエラーに関しては,ユーザに対処方法を指示できるようにしたい.

謝辞

本研究は JSPS 科研費 JP16K00463 の助成を受けたものです.

参考文献

[1] 新井侑太, 鈴木徹也, 相場亮. 手書き変体仮名認識に おける制約充足問題の拡張. 第 75 回情報処理学会全国大会講演論文集, pp 331 - 332, 2013.

[2] 渡辺悟, 鈴木徹也, 相場亮. 手書き変体仮名認識における単語の連接関係を用いた解の絞り込み. 情報処理学会論文誌, 56 巻 3 号, pp 951 - 959, 2015.

[3] 山本純子, 大澤留次郎. 古典籍翻刻の省力化：くずし字を含む新方式 OCR 技術の開発. 情報管理, Vol.58, No.11, pp 819 - 827, 2016.

[4]国立国語研究所. 中古和文 Unidic ver1.3. http://www2.ninjal.ac.jp/lrc/index.php?UniDic%2F%C3%E6%B8%C5%CF%C2%CA%B8UniDic