自然言語処理適用のための OCR 後処理技術の提案

鈴木　敏　　永田　昌明
NTT　コミュニケーション科学基礎研究所
{suzuki.s, nagata.masaaki}@lab.ntt.co.jp

1　はじめに

近年の OCR 技術の進歩により、文字認識の精度は向上している。例えば、PDF ファイルのような綺麗な文字画像の処理では、かなりの高精度で文字認識が可能となった。しかしながら、看板の写真等、実世界の文字列や背景のある文字、挿絵入りの本の処理などでは誤認識が多く、いまだ十分な精度があるとは言えない。

これに対し、自然言語処理技術の適用により誤りを訂正しようという検討もなされている [7, 5]。このような技術の導入により文字認識精度が向上することが期待できる。ところが、実際にこれらを利用しようとすると、自然言語処理側では対応できないほど質の悪い低レベルな OCR 出力を得ることがある。

このような問題に対処するためには、OCR 側の技術の向上が不可欠ではあるが、長い歴史の OCR 研究の成果が現状であることを考えると一朝一夕に技術的な解決ができることは期待できない。そこで本稿では、OCR の後処理的な機能を付加することにより、OCR の出力を自然言語処理が適用できるレベルにまで向上させることを試みる。

一般に、OCR の出力は最終的な文字列を出すように設定されている。すなわち、その出力を言語処理的に処理することを前提とはせず、最終的な結果として出力を処理するため、余分な情報は切り捨てている。ところが、自然言語処理にとってはこれらの余分な情報が有用な場合もある。また、不確かな文字情報から正しい文字列を推定するのは、自然言語処理にとって得意とするところである。

このような技術的な背景を鑑み、本稿で提案する技術は、OCR からできる限り多くの情報を取り出す手法である。本手法では、多くのゴミを含んだ出力を出すことを厭わず、その中に少しでも多くの正しい文字が含まれることを目的とする。言い換えれば、精度（適合率）ではなく、再現率を高めるための手法である。

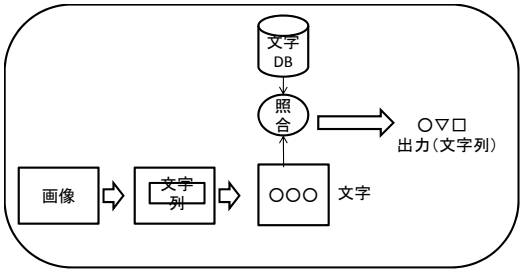

図 1: 一般的な OCR 処理

2　提案手法概要

OCR による印刷文字の認識では、文字認識誤りの原因は、文字位置指定の誤りにあることが多い。そこで、本稿で提案する手法は、OCR 出力から文字位置を再推定する手法である。

図 1 に示すように、一般的な OCR では、画像を与えられると文字（あるいは文字列）位置を特定し、その範囲内で取り出した文字画像と、文字 DB にある文字情報とを照合し、一致度の高い文字を出力とする [4]。照合では数値化された識別距離を用いるのが一般的である。また、言語処理機能を付加する場合は、この出力に対し処理を行う。

提案する手法は、取り出した文字出力の中から信頼度が高い文字を選別し、これらを基に、画像上にある文字の位置を再推定する手法である。図 2 にその概要を示す。

提案手法では、まず、照合で得られた文字に対して信頼度判定を行う。信頼度判定は照合時に利用する識別距離を利用し、閾値を設定して判断する。また、文字サイズが大きく異なるものは除外する。更に、例えば日本語であれば文字コードから判断するなど、言語的なフィルターも利用する。

次に、このようにして得られた文字を正しく認識できた文字と仮定し、これらの文字の位置を基準に入力画像全体に等間隔に文字が広がっていると仮定して、

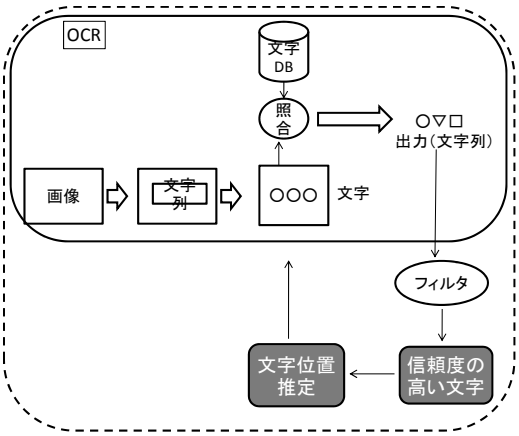

図 2: 提案手法のモデル

文字位置を推定する。

この推定した文字位置を OCR に与え、文字を再推定することで、初回の推定では認識できなかった文字を得られる可能性がある。また、上記処理を繰り返すことで、更に新たな文字を認識できることも考えられる。

以上の手順をまとめると、

1. OCR の出力文字を文字位置、文字サイズ、識別距離と共に取り出す。
2. 識別距離の閾値、文字サイズの最頻値、文字コード等を利用し、フィルターリング。
3. 残った文字の文字位置と文字サイズから、入力画像全体に広がるように文字位置を推定。
4. 推定文字位置を OCR に戻し、文字の再推定。
5. 1 ～ 4 を数回繰り返す。

となる。

初回の OCR 出力も含めて、各回の文字推定結果の全てを最終出力として扱う。このとき、最終的に取り出される文字数は繰り返し回数に従い増加し、ゴミも増えることになるが、当初の目的である再現率の上昇も期待できる。

3 実施例

提案手法を実際に適用した例を次に示す。OCR ライブラリーとしては、パナソニックソリューションテクノロジー社の製品を利用した [6]。

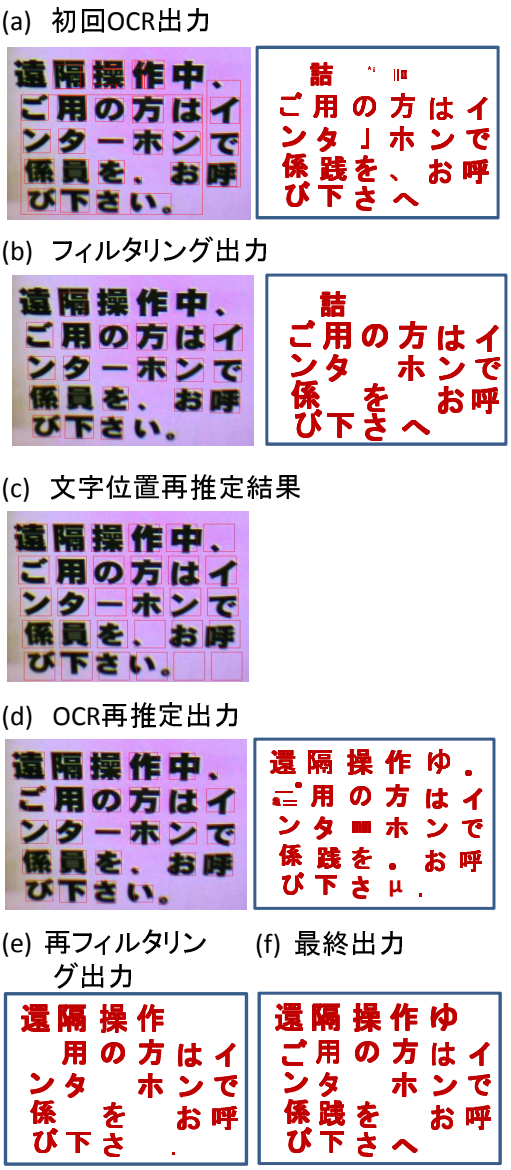

図 3: 実施例

図 3 は実際の看板の写真画像を対象に、提案手法を適用した結果である。(a) は提案手法を適用する前の、初回の OCR 出力である。画像上には文字認識時の文字位置とサイズを四角で重ねて表示してある。認識結果は、2 行目から 5 行目までは大まかに文字を取り出せており、言語処理を通せば正しい文字列を推定できるレベルである。しかしながら、1 行目は正しい文字を全く取り出せておらず、言語処理を通しても文字列を推定できるレベルにはない。

(b) は初回の OCR 出力に対し、識別距離、文字コード、文字サイズを利用してフィルタリングを行った結果である。識別距離は 900 以下 [6]、文字種は日本語に限定した。文字サイズは初期 OCR 結果の文字サイ

ズの最大値と最小値の間を 5 等分し、クラス毎に頻度を取り、最大頻度のクラスのサイズ± 20%を適合文字サイズとした。この結果、記号などの文字は除外された出力が得られている。これらの文字は信頼度が高いと考えられる。

(c) は文字位置を再推定した結果である。推定された文字位置が画像上に四角で表されている。文字位置の推定方法は、次のとおりである。

1. (b) で取り出した信頼度の高い文字の位置を、水平方向、垂直方向それぞれで近いもの同士でまとめる。例えば、水平方向であれば、「ご」「用」「の」「方」「は」「イ」が一つの組になる。
2. 次に、これらの文字位置の中心を通る直線を考える。各文字の中心位置に対して線形回帰により直線を決定する。
3. 全ての垂直方向、水平方向の文字の組に対し、同様に直線を決定する。
4. 画像全体のサイズと文字サイズと直線の間隔と比較し、直線が不足する部分に直線を補足して加える。
5. 各直線の交点を文字の中心とし、直線の間隔を文字サイズの上限として文字位置が重ならないように調整し、各文字位置を推定する。

このように文字位置を推定することにより、文字が取り出せなかった位置を含めて文字の再推定を行うことが可能になる。また、線形回帰を利用することで、画像の歪みにも対応できる。図 (c) を見ると、再推定のための文字位置が等間隔に、画面全体に広がっていることがわかる。

(d) には文字の再推定結果が示されている。再推定では、(c) で推定された文字位置・サイズの範囲内で文字を再推定しており、(c) で得られた文字枠よりも文字にフィットした文字枠が示されている。認識結果として取り出された文字からは、初回 OCR 結果 (a) では取り出せなかった 1 行目の文字を大まかに認識できていることがわかる。逆に、「ご」の文字に注目すると、(a) では正しく認識できていたが、再推定では複数の文字に分解して文字推定をしており、正しく認識できていない。

(e) は (d) の結果を再度フィルタリングしたものである。フィルタリングの条件は (b) で行ったものと同じである。

(f) は初回の OCR 結果 (a) と再推定結果 (d) とを統合した結果を示している。結果を統合するにあたり、

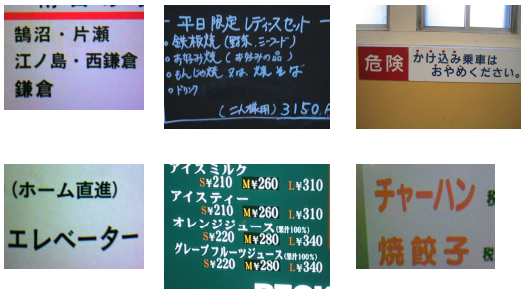

図 4: テスト画像サンプル

文字位置が重なる文字は識別距離により選択するという方法を用いている。

上記手順により取り出された文字列は不完全ではあるが、言語処理を通せば正しく推定できるレベルである。

4 再現率の評価

前述したように、提案手法の目的は再現率を上げることにある。そこで、1000 枚の実画像および 3 冊の絵本（見開き合計 35 枚）[1, 2, 3] を対象に、初回 OCR の結果と再現率を比較した。

1,000 枚の実画像は全て看板を写したもので、駅の看板から店のメニューに至るまで様々なものが集められており、字体も手書きを含めて様々である。例をを図 4 に示す。一方、3 冊の絵本は、絵と文字が分かれているものから完全に重なっているものまで、様々である。

看板画像と絵本を比較すると、看板は文字数は少ないが字体、文字サイズが統一されていないものが多い。一方の絵本は文字数は多いが、字体、文字サイズは統一されている。また、絵本には画像全体に対する文字の割合が小さいという特徴もある。

再現率 (p) の計算は、正解文字が推定結果中に現れる確率で、正しい文字の数 (x)、推定文字中の正解文字数 (y) を使って、

$$p = \frac{x}{y} \tag{1}$$

で表せる。

看板画像および絵本の文字の再現率の計算結果は表 1 に示す通りである。

看板画像では、劇的な変化は無いが、僅かに再現率は上がっている。実験で用いた 1000 枚の画像の中には初回 OCR でほぼ全ての文字を認識できるものも多

	初回 OCR	提案手法
看板画像	22.33%	22.61%
絵本	15.39%	28.09%

表 1: 再現率の比較結果

く、また、手書き文字の看板のように提案手法を適用しても全く認識できない画像も多く含まれているためと考えられる。

一方、絵本での結果は再現率がほぼ 2 倍になっており、提案手法が有効に働いていることがわかる。絵本では、絵が邪魔になり文字位置の特定が難しいが、文字は字体もサイズも統一されている。そのため、文字位置さえ正しく推定できれば文字認識が容易である場合が多いと考えられる。

これを示すのが図**??**である。全ての画像に対し、正解文字数を、横軸に初回 OCR での文字数、縦軸に提案手法での文字数として画像毎にプロットしている。ほとんどの画像は正解文字数が同数だが、一部の画像では非常に多くの文字を取り出せている。

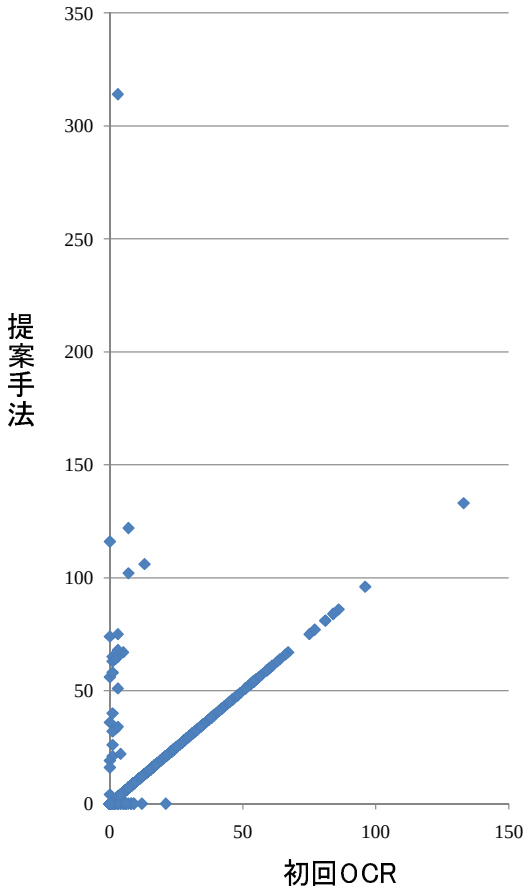

図 5: 正解文字数の散布図

5 おわりに

OCR の結果を言語処理へ渡すことを前提とした、OCR 事後処理的な手法を提案し、簡単な評価実験を行った。看板画像および絵本による検証を行った結果、絵本のように文字自体は認識しやすいが文字位置を特定しにくい画像で特に有効であることがわかった。ただし、看板画像では効果が得られないと言うことではなく、実施例に示したように、有効に機能する場合もある。

本手法のデメリットとしては、計算量とゴミ出力が増えるという問題がある。計算量は文字数や繰り返し回数に従い大きくなる。特に、画像全体に対して一文字の大きさが非常に小さい絵本等では 1 回の再推定でも計算量はかなり大きくなってしまう。ゴミ出力が増える問題も含めて改善の余地はある。

今後はこれらの課題への対応と、後に続く自然言語処理技術となる誤り訂正手法を検討する

参考文献

[1] いないいないばあ. 童心社, 1967.

[2] かたつむりののんちゃん. 童心社, 1999.

[3] しろかぶくんとアンパンマン. フレーベル館, 2011.

[4] Ray Smith. An overview of the tesseract ocr engine. http://tesseract-ocr.googlecode.com/svn/trunk/doc/tesseracticdar2007.pdf.

[5] Takafumi Yamazoe, Minoru Etoh, Takeshi Yoshimura, and Kousuke Tsujino. Hypothesis preservation approach to scene text recognition with weighted finite-state transducer. *ICDAR*, 2011.

[6] パナソニックソリューションテクノロジー. 活字認識ライブラリー ver.13. http://panasonic.biz/it/sol/ocr/sdk/textocr/index.html.

[7] 永田昌明. 文字類似度と統計的言語モデルを用いた日本語文字認識誤り訂正法. 電子情報通信学会論文誌 (D-II), Vol. J81-D-II, No. 11, pp. 2624–2634, 11 1998.