1G3-04

ドキュメント画像から文書検索を行うための XML 定義の提案

The proposal for XML definition to retrieve documents from the document image

松本 馨[*1]
Kaoru MATSUMOTO

櫻田 武嗣, 中川 正樹[*2]
Takeshi SAKURADA, Masaki NAKAGAWA

[*1] 学校法人産業能率大学 総合研究所
Research Institute, The Sanno Institute of Management

[*2] 東京農工大学工学部
Tokyo University of Agriculture and Technology

This paper describes XML definition to retrieve documents from document images. It has been a problem that document management systems which use Optical Character Recognition (OCR) often pose some system requirements and do not avail specifications of the data, although many products have already been invested and used. Moreover, they assume entire translation from document images to codes so that verification of OCR has been a heavy burden for operators. This paper propose an XML definition, which provides high compatibility of the data and keeps multiple OCR candidates to dispense with the verification labor of OCR.

1. はじめに

近年, 共通の仕様を定めることでデータの相互運用性を高め, 一般に普及させるための XML 形式が多く見られるようになってきた. OCR を使用したドキュメント管理システムは, 既に多くの製品が存在し, 運用されている. しかし, これらは動作する環境が限られていることや, データの詳細仕様が明らかでないこと, そのデータを利用した新たなアプリケーション開発が第三者には困難であることが問題である. また, 画像から文字列への全置き換えを前提としているものが多く, 認識結果を確認・訂正する作業が作業者にとって大きな負担となっている. そこで, ドキュメント画像の OCR 結果を仕様の明らかな XML 形式で保存し, それを文書検索に利用するための XML 定義の提案を行う.

2. 背景

電子文書の分野では Adobe 社の pdf 形式が最も有名で, よく用いられているのはいうまでもない. これは, 多くの OS 環境で動作し仕様も公開されており[1], 閲覧ソフトが無料で配布されていることから, 広く用いられている. しかし, pdf 形式は, 文字コードや文字フォント, 図表などの書式を保存し, 他環境でも元データに近い体裁で読めるようにすることを目的としており, 画像として取り込まれたデータの保存にはあまり向いていない.

pdf 形式は, 透明テキストとして画像に文字情報を付加することが可能である. しかし, これは使用する画像形式が固定的で, ほとんどユーザーが選択できず, 既に存在する多様な画像ファイル形式に対応できない. また, 仕様が公開されているとはいえ, pdf 形式自体の利用が難しく, これに対応した検索機能を持つソフトウェアは多くない. 一部のファイル検索用ソフトウェアや google をはじめとする Web 検索エンジンがサポートしている程度で, 個人で手軽に利用できるものとは言い難い状況である. 加えて, ユーザー主導でないバージョンアップにより, ソフトウェアのバージョンに依存した問題が発生する可能性も残っている. 他にも文書画像管理のためのツールはいくつか存在するが, これらも同様の問題を抱えている.

このため, 既存の画像形式(jpeg や png など)から独立した付加情報として, 仕様の明らかな XML 形式を用いたメタデータを文字情報として付加することが有効であると考えた. これは, 既に存在する画像閲覧／管理ソフトウェアが利用可能であることや, 今後, 新しい画像形式が登場しても, 互換性が維持されたり, 画像変換用ソフトウェアの登場が見込まれるからである. XML により付加されるメタデータと画像ファイルが分離していることで, 画像形式の変換にも容易に対応可能となり, 付加情報のハンドリングも容易に行うことが可能となる.

連絡先:松本 馨, 学校法人産業能率大学 総合研究所,
〒158-8630 東京都世田谷区等々力 6 丁目 39 番 15 号,
MATSUMOTO_Kaoru@hj.sanno.ac.jp

3. 目的

3.1 文字認識システムの概要

文字認識システムは, 大別するとペンタブレット上で筆記情報をリアルタイムに取得するオンライン文字認識と, 紙や画像として残された筆記情報(画像情報)をもとに文字認識処理を行うオフライン文字認識の 2 種類がある. 本研究は, オフライン文字認識を用いるものである.

オフライン文字認識システムは, 大きく分けると前処理と文字認識処理, 後処理の 3 つの処理で構成される(図 1). このうち, 前処理では, 文字認識エンジンにかける前の画像処理(グレースケール化, 2 値化処理など), 行切り出し, 文字切り出し, 正規化処理を行う. 文字認識処理では, 切り出した文字画像に対して統計的手法による文字認識処理を行い, 辞書に登録された文字情報との距離計算を行い, 候補文字列を算出する. 後処理では, 文脈による処理を行い, 候補文字列とその距離値から一部の認識結果の入れ替えを行い, より文章として妥当な文字列になるように認識結果の修正を行う. 多くの OCR システムでは, この距離値による候補文字の確からしさが一定値以下の場合に, 誤認識の可能性が高いパターンとして, 作業者に認識結果の確認を促し, 目視による修正を行っている.

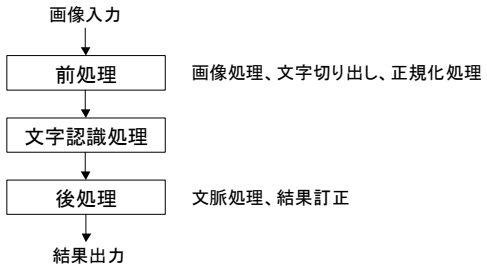

図 1 文字認識システムの構成

3.2 文字認識システムの問題点

文字認識システムを利用するにあたって問題となる点として下記が挙げられる.

(a) 画像の 2 値化処理
隣接する文字がつながらなく, かつ, 1 つの文字が細切れにならないようにグレースケール画像を 2 値画像に変換すること

(b) レイアウト認識
文字のレイアウト(縦／横書き, 段組など)を認識し, 1 つの文章としてつながるようにすること

(c) 文字切り出し
文字画像を 1 文字ずつ切り出して, 文字認識エンジンに渡すこと

(d) 認識結果の決定と修正
候補文字列(認識スコアの上位 1 位～10 位程度を使用)の中から妥当なものを選択すること

この中で(b)は非常に難しい問題であるが, 認識結果を文章として扱うのではなく, 画像上の座標に文字を対応付ける場合, 大きな問題ではなくなる. (a),(c)も重要な問題であるが, 利用者が実際に介入できるようにしているシステムは少ない.

利用者にとって大きな負担となっているのは(d)である. 認識結果は, 基本的に 1 位候補を結果として用いることが多いが, 認識スコアが 1 位候補と他の候補で大きく変わらない場合, 文脈後処理により順位の入れ替えが行われ, 2 位以下の候補が認識結果として採用されることがある. また, スコアが悪く, 結果が妥当でないと判断される場合, 利用者に認識結果が正しいかどうか確認を求める処理を入れることで, 認識結果訂正を行うことがある.

認識結果訂正については, 認識結果を 1 つに特定しないことで訂正を不要にすることができると考える. これは, 文字認識システムの認識率を考えるときに, 1 位認識率について着目すると必ずしも認識率が高くないが, 例えば 10 位認識率(1～10 位候補までに正解が含まれる確率)で考えれば, その認識率は, より 100%に近づくからである[2]. つまり, 1 つの文字に複数の候補文字を付加しておくことで, その中のどれかに正解が含まれている状態にするのである.

この方法は, 文字認識システムに対する負担を減らす代わりに, その後の利用で負担が増える危険がある. つまり, 本来の正しい文字と関係のない情報が付加されていることで, 検索の効率が落ちてしまう危険である.

しかし, 実際に検索を行う場面では, 複数文字で構成された単語を用いる場合がほとんどであり, 1 文字での検索を行うのはそれほど多くないと思われる. つまり, 2 文字以上の組み合わせで検索をかける場合, 多少, 間違った文字が含まれていても, その間違った文字が組み合わさった形でその検索キーワードと一致する確率は低いであろうと考えられるのである.

3.3 本定義の目的

本定義では, XML 形式を用いて次に示す情報(OCR による文字認識結果)を記録し, これをもとに画像ファイルにどのような文字が書かれているか検索できるようにすることを目的とした.

(a) 候補文字列情報
(b) 文字認識スコア情報
(c) 文字の画像上の座標情報
(d) 画像ファイルの場所情報
(e) 文字認識エンジン名称, バージョン情報

通常, OCR では文章化したものをデータとして保存するが, そのためには認識率を 100%に近づける必要があり, 加えて, レイアウト認識や文字列切り出しを正確に行わせる必要がある. これには多大なコストや手間がかかるため, OCR 結果は検索のみに使用し, 元画像データは破棄しないで併用する設計とする. つまり, データの表示は元々の画像データを表示し, 文字列検索をかける時は OCR 結果を利用する.

このような検索方式は, 既に一部で試行されている. 例えば, 目録カードの検索である[3]. 目録カードは, 一定のサイズ, 一定の書式で書かれているため, OCR による処理に向いている. しかし, 万単位の数量ある目録カードを完全に文字コードに置き換えるのは, 内容確認, 修正の手間を考えると現実的ではない. かといって, 単に画像情報としてデータを残すだけでは省スペース性や, 同時に多人数が閲覧できるといったメリットはあるものの, 検索性向上に活かされない.

このため, 画像情報はそのまま残しておき, 文字列検索を行う部分に OCR 結果を使用することで, 画像を文字に置き換える手間をなくし, 認識結果の間違いによる情報の損失を防ぐことが可能になると考えた.

4. XML 定義

4.1 定義項目

ここでは, OCR 結果を文書検索に使用するための XML 定義を考える. まず, 使用する項目を次の通りに定めた.

(a) DocOcr 定義
Id:識別子
Date:データを出力した日時
SystemName:OCR システムの名称
SystemVersion:OCR システムのバージョン情報
Image:画像データへのリンク情報

(b) OcrResult 定義
LocationX:画像上の文字の x 座標
LocationY:画像上の文字の y 座標
SizeX:文字画像の横幅
SizeY:文字画像の縦幅
CandidateNumber:候補文字数
CandidateString:候補文字列
CandidateScore:候補文字スコア

DocOcr 定義では, 文字認識システムについての定義と, 認識にかけた画像データの情報を格納する. ここで, OCR システムの名称やバージョン情報を格納するのは, 文字認識システムによって出力される結果の傾向が異なる可能性があるからである. 例えば, 候補文字のスコアなどは, 文字認識システムによって出てくる値の範囲が異なったり, 値が小さい方が正解に近いのか, 遠いのか, などが異なるのである.

OcrResult 定義では, 1 文字ごとの文字認識結果を格納する. ここでは, Location で画像のどこにその文字が書かれているかの座標(左上を原点とした x, y 座標系)を記録し, Size でその文

字の大きさ(x, y で横幅, 縦幅を示す)を記録する. さらに, 認識システムが出力する文字候補の数(通常, 10 程度まで使われる)と, その文字候補(通常, 1～10 位候補程度まで使われる)と, それに対応するスコアを記録する.

実際の使用にあたっては, DocOcr の集合である DocOcrs を定義し, 各 DocOcr(1 つの画像ファイル)の中に含まれる 1 文字ごとに OcrResult が記述される形式になる.

4.2 具体例

前項で述べた定義をもとに, 実際の画像(図 2)に対してタグ付けを行った例を図 3 に示す.

この例では, 画像の左上の「現在、日」の部分を認識した結果を示している.

ここでは 3 文字目を誤認識し, 認識結果が「ヽ」となっているが, 2 位候補として正しい認識結果である「、」が含まれていることが分かる. ここでのスコアは「ヽ」が 231.4, 「、」が 261.9 であり, 1 位候補の方がスコアが小さい. スコアが小さいものほど確からしさが高くなるというのは違和感があるかもしれないが, この数値は文字認識システムの辞書にある代表パターンとの距離値を示しているため, 値が小さいものほどその代表パターンに近い(完全一致すれば距離値は 0 になる)ことを示しているからである. これは, OCR システムに依存する違いなので, これらの値の見方は SystemName に書かれた名前をもとに判断することになる.

4.3 検索方法

キーワードによる検索は, 基本的に複数文字を用いて行う. 1 文字のキーワードでは, 大量の文字がヒットしてしまい実用にならないと思われる. これは, インターネットの Web 検索でも同様だが, なるべく複数のキーワードを用いる方がヒット率が上がると思われる.

この定義では文字を文章として格納しているのではなく, 単に画像上の座標として位置情報のみを記録している. つまり, 検索に使う場合, 複数の文字同士が近い位置にあるかどうかで判断することになる.

この検索方法が妥当であるかどうかは, 実際の画像にタグを割り当てて, それをもとに検索を行い, 探している情報がヒットするかどうか, ヒットした場合, 目的のものを見つけ出すまでにどれだけの労力を必要としたか(例えば, 関係ない情報が出た回数)などから妥当性の評価基準を作り, 評価を行う必要があると考える.

5. おわりに

本稿では, ドキュメント画像に対して OCR 処理を行った結果をメタデータとして格納し, これを検索に用いるための XML 定義について述べた. 画像ファイルとメタデータを独立させることで, 画像ファイルの運用が容易になる. さらに, テキストで記述されたメタデータは第三者による検索アプリケーション開発が容易になると考える. 検索方法の妥当性については, 今後, 実際のデータに適用し, 妥当性の評価基準を定め, それをもとに妥当性判断を行う必要があると思われる.

参考文献

[1]http://partners.adobe.com/asn/tech/pdf/index.jsp

[2]O.Velek, S.Jaeger, M.Nakagawa,"A New Warping Technique for Normalizing Likelihoods of Multiple Classifiers and its Effectiveness in Combined on-line/off-line Japanese Character Recognition," Proc. 8th IWFHR, pp.177-182, Niagara-on-the Lake, Aug. 2002.

[3]A.C.Downton, A.C.Tams,G.J.Wells, A.C.Holmes, S.M.Lucas, "Constructing Web-Based Legacy Index Card Archives - Architectural Design Issues and Initial Data Acquisition," Proc. 6th ICDAR, pp.854-858, Seattle, Sept. 2001.

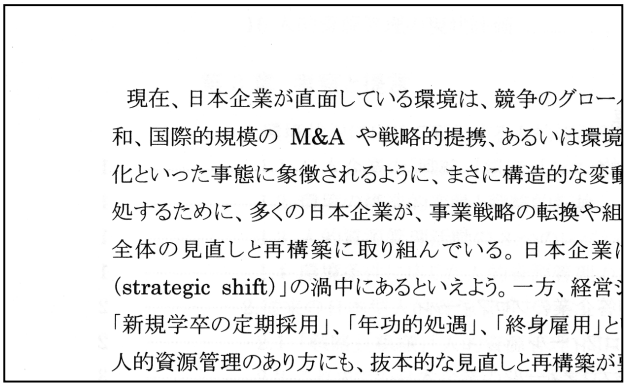

図 2 入力画像例(一部)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<DocOcrs>

  <DocOcr  Id="ocr_200404190801"
    Date="2004-04-19"
    SystemName="HANDS-OCR"
    SystemVersion="1.00"
    Image="./sample01.png">

    <OcrResult  LocationX = "287"
      LocationY = "592"
      SizeX = "39"
      SizeY = "46"
      CandidateNumber = "5"
      CandidateString = "現,硯,規,視,院"
      CandidateScore = "464.4,516.6,531.4,621.2,686.9"/>

    <OcrResult  LocationX = "331"
      LocationY = "592"
      SizeX = "39"
      SizeY = "46"
      CandidateNum = "5"
      CandidateString = "在,庄,荏,往,左"
      CandidateScore = "388.4,532.1,542.2,551.5,558.0"/>

    <OcrResult  LocationX = "380"
      LocationY = "592"
      SizeX = "7"
      SizeY = "46"
      CandidateNum = "5"
      CandidateString = "ヽ,、, ,ゝ,・,.  "
      CandidateScore = "231.4,261.9,298.8,307.2,327.6"/>

    <OcrResult  LocationX = "415"
      LocationY = "592"
      SizeX = "22"
      SizeY = "46"
      CandidateNum = "5"
      CandidateString = "日,曰,目,且,白"
      CandidateScore = "291.1,314.8,367.6,431.1,434.7"/>

    ・・・・・・
    ・・・・

  </DocOcr>
</DocOcrs>
```

図 3 結果出力例