OCR による文書検索を備えた Python ベースの文書管理システムの開発

中田　光一†　　浦山　康洋†
† 高知工業高等専門学校ソーシャルデザイン工学科

1. はじめに

近年，コスト削減や業務効率化を目的とした紙文書の電子化が企業等で推進されている．一方で，紙文書をただスキャンしてファイルサーバに保存するだけでは必要な資料を探しだすために長時間を要してしまい，業務効率がかえって落ちてしまう．上記の課題を解決する方法の一つとして文書管理システムを導入することが考えられるが，当該システムは導入コストおよび維持費が高いこともあり，いまだ紙文書の電子化に踏み切れていない組織も多々ある．

そこで，本稿では文書管理システムを安価に構築することを目的とする．具体的には，OCR（Optical Character Recognition/Reader：光学文字認識）による文書内検索機能に加え，電子文書を閲覧・管理できるユーザインタフェースを Python で実装する．

2. Python による文書管理システムの実装

図 1 に作成した文書管理システムのホーム画面を示す．本画面は Python 用の Web アプリケーションフレームワークである Flask を使って作成した．画面中のテキストボックスに検索ワードを入力して「検索」ボタンをクリックすると Python プログラムが動作し，システム内に保管されているすべての電子文書に対して OCR を使った全文検索が行われる．その後，検索結果が図 1 の下部分に表示される仕組みとなっている．

文書内検索を行う Python プログラムの処理手順を以下にまとめる．本プログラムを実装するにあたり，今回はオープンソースの OCR エンジンである Tesseract を採用した．

① 文書ファイル（PDF 形式を想定）を画像ファイルに変換する．この変換は pdf2image を用いて行っており，PDF ファイル 1 ページごとに画像を 1 枚生成している．
② ①で得られた画像群に対し Tesseract による OCR を実行し，画像ファイルから文字列を抽出する．
③ ②で抽出した文字列と検索ワードを比較し，文書内に検索ワードが何個含まれていたかをカウントする．
④ システム内すべての文書ファイルに対して①～③を実行し，検索ワードを含むファイルをすべて列挙する．

3. 性能評価とプログラムの改善

図 2 は作成した文書管理システムの検索速度に関する調査結果を示している．なお，本調査は CPU が Intel Core i7-8565U，RAM が 12GB であるノート PC を用いて行って

図1．文書管理システムのホーム画面

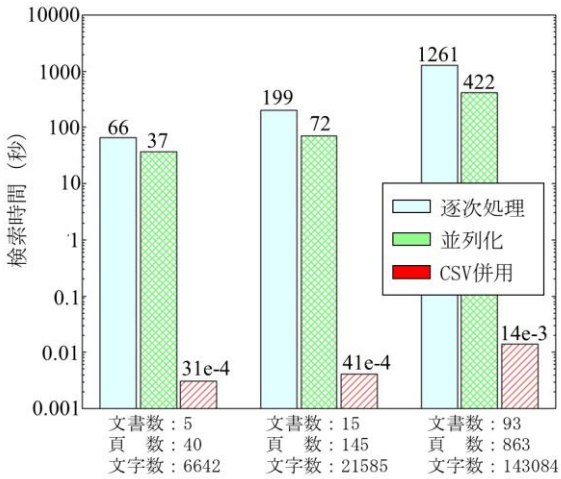

図 2．システムの検索時間

いる．最初に作成したプログラム（逐次処理）では，文書数が 5 であったとしても検索に 66 秒もの時間を有してしまうことがわかった．そこで，検索時間を短縮するためにプログラムの並列化を行った．これにより，逐次プログラムと比べて検索時間を短縮することに成功した．

しかしながら，並列化プログラムの検索時間も決して早くはないことが図 2 からわかる．そこで，電子文書をアップロードしたタイミングで OCR の処理を行い，結果を CSV 形式で保存するようプログラムを改良した．これにより，検索時間を大幅短縮することができた．

4. まとめと今後の展望

本稿では OCR と Python を使った安価な文書管理ステムについて構築状況を報告した．今後はより多くの電子文書を使ってシステムの性能を調査し，当該システムの実用性の評価とさらなる機能改善を行う予定である．