

タブレット端末によるカメラ機能を用いた学術論文閲覧支援の一手法

谷尻 淳喜[†] 太田 学[†] 高須 淳宏^{††} 安達 淳^{††}

[†] 岡山大学大学院自然科学研究科 〒700-8530 岡山県岡山市北区津島中 3-1-1

^{††} 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: [†]{tanijiri, ohta}@de.cs.okayama-u.ac.jp, ^{††}{takasu, adachi}@nii.ac.jp

あらまし 非専門家が学術論文のような専門性の高い文書を読む場合、未知の語に遭遇する可能性が高く、内容を理解するのに時間がかかる。そのため論文中の専門用語等の重要語を予め自動抽出してタブレット端末のユーザに提示する学術論文閲覧支援インタフェースが開発されている。本稿では、タブレット端末のカメラ機能を用いて、紙媒体の学術論文の閲覧支援を行うインタフェースを提案する。提案インタフェースでは、iPad のカメラ機能により紙媒体の論文からテキストをリアルタイムに検出する。また、検出したテキストに対して解析結果や Web 上の関連情報を論文画像上に表示する。評価実験では、提案インタフェースと前野のマルチインタフェースと iBooks を使って、設定した課題を終えるまでの時間とインタラクション数を計測し比較した。その結果、課題を終えるまでにかかった時間は iBooks が最も短い結果となったが、インタラクション数は提案インタフェースが最も少ない結果となった。

キーワード 重要語, 閲覧支援, 学術論文閲覧支援, OCR

1. はじめに

近年のタブレット端末や電子書籍閲覧端末の普及により、従来紙媒体で読んでいた文書をタブレット端末等を用いて読む機会が増加した。この読書形態の変化に伴い、電子媒体の閲覧支援に関する研究が行われている。

前野ら [1-3] は英語の学術論文 PDF を対象に、タブレット端末による学術論文閲覧支援のためのインタフェースを開発した。前野らのインタフェースでは、テキストの埋め込まれた学術論文 PDF ファイルを用い、1 単語または 2 単語の単語列に対して、単語列の重要度等の論文の解析結果や Web 上の関連情報をユーザに提示する。また、論文中の重要語として省略語の一つである頭字語に注目し、論文中出现する頭字語とその実体語の抽出手法を提案した。これにより、頭字語を検索するとき、実体語の曖昧性が解消され、ユーザはスムーズに解説ページにたどり着くことができる。

学術論文のように専門性の高い文書を読む場合、未知の専門用語等に遭遇する可能性が高く、その度に辞書を引く、Web サイトで検索するといった行為は効率が悪い。前野らの学術論文閲覧支援インタフェースは有用である。しかし、学術論文の PDF ファイルがあっても、思いついたことを手軽に書き込めたり、前後の文脈を自由に行き来できたりするといった利点から、紙媒体で閲覧することも多い。そこで本稿では、タブレット端末のカメラ機能を用いて、紙媒体の学術論文の閲覧支援を行うインタフェースを提案する。

提案インタフェースでは、iPad 等のタブレット端末のカメラ機能を用いて紙媒体の論文を撮影し、論文画像を取得する。取得した画像中のテキストの単語列の重要度を算出し、ユーザに重要度順に提示する。また、抽出した単語列に対して重要度等の論文の解析結果や Web 上の関連情報を表示する。本インタフェースによる支援対象ユーザは、その分野の初学者を想定し

ている。また、提案インタフェースは iPad アプリケーションとして実装し、タップやスワイプ等の直感的な操作が可能である。

以下に本稿の構成を示す。2 節で本研究の関連研究を紹介し、3 節で提案インタフェースの実装について述べる。4 節で実装した提案インタフェースによる学術論文閲覧支援について述べ、5 節で評価実験について説明する。6 節で本稿をまとめ、今後の課題について述べる。

2. 関連研究

2.1 紙媒体と電子媒体の比較

柴田ら [4] は、娯楽を目的とした読み方では、ページめくりが頻発しない限り、タブレット端末と紙媒体では認知負荷に大きな違いはないことを報告した。しかし、答えを探すことを目的とした業務での読み方では、答えのあるページがどこにあるかあいまいな場合、電子媒体ではページの行き来に時間を要した。一方紙媒体では、指を挟んで特定のページにすぐに移動することができたり、ばらばらとページをめくることができたりするため、紙媒体のほうが早く答えに辿り着くことができたことが報告されている。

小林ら [5] は、国私立大学 5 校の学部生及び大学院生 36 名の被験者に対して、タブレット端末と紙媒体のそれぞれを用いて、文学的文章と説明的文章の 2 種類の文章を読ませ、主観評価および読み速度、記憶テスト、理解テストを行った。その結果、被験者の主観的な読みやすさでは、iPad は紙と同等の性能を実現していることが確認された。しかし、文章理解ではタブレット端末よりも紙媒体に優位性があったと報告されている。

2.2 論文閲覧支援システム

阿辺川ら [6] は、脚注表示機能を備えた論文閲覧システム SideNoter を開発した。このシステムは Web ブラウザ上で動作し、表示されている論文の左右の脚注部に、Wikipedia をリソースとした補足情報を表示する論文読解支援機能を有してい

る。また、本文検索、専門用語の Web 検索、連続ページめくり、書き込み機能等の論文閲覧機能も有しており、紙や PDF 等の一般的な論文フォーマットと、開発したシステムの機能特性を比較した。

川上ら [7,8] は、論文とプレゼンテーションスライドの自動対応付けを行う手法を提案した。このシステムでは、論文とその論文を説明するプレゼンテーションスライドの対を入力とし、1 枚のスライドに連続する 1 段落毎の paragraph 粒度で区切られた文章の集合を対応付ける。対応付けには、まずスライドのタイトルと論文中のセクションタイトルのコサイン類似度等を用いて軸となるスライドを決定する。次に、各スライドとセクション内の paragraph との内容類似度を算出し、内容類似度をスコアとして各 paragraph をランキングして判定する。

鉢木ら [9,10] は、OCR テキストを用いた学術論文閲覧支援システムを開発した。具体的には、論文のタイトル、概要、キーワード部分から専門用語を抽出し、それらの語についての解説やツールなどの有用なページへのリンクを提供した [9]。また、抽出した各専門用語で検索される論文集合と、それらに出現する専門用語集合間にリンクを生成し、この二部グラフに HITS アルゴリズムを適用することで、関連論文をランキングして推薦するシステムを提案した [10]。

2.3 重要語抽出

今井ら [11] は、係り受け関係と共起関係を導入した文書グラフを構築する方法を提案し、これを解析することで文書の特徴ベクトルを抽出し、重要語を抽出した。構文解析した文から文節間の係り受け関係を抽出し、さらに共起関係を考慮することで文書グラフを構築した。重要語の抽出実験において、提案した特徴ベクトルによる重要語抽出手法に IDF を付加したものは TF-IDF と比較して高い精度を示した。

小倉ら [12] は、近年の莫大なテキストデータを扱う問題には、自動でカテゴリごとに分類できるような文書分類手法が必要であると考えた。そこで、PageRank を用いた重要語の抽出を行い、それに基づいて重要文を抽出し、潜在的意味によるクラスタリングを行う手法を提案した。重要語の抽出に PageRank、または TF-IDF を使い、k-means 法を用いてクラスタリングを行った結果、PageRank の方が文書分類の精度が向上したと報告されている。

3. 提案インタフェースの実装

3.1 概要

提案インタフェースは、紙媒体の英語の学術論文の閲覧支援のためのインタフェースである。本インタフェースはタブレット端末のカメラ機能を用いて論文画像を取得し、画像中のテキストを利用して、論文を閲覧する際に有用となる Web 上の解説ページ等を提示する。本インタフェースは表 1 の環境下で開発した。

3.2 タッチ操作

本論文で用いるタッチ操作を表 2 にまとめる。提案するインタフェースは iOS のアプリケーションであるため、スワイプ等のタッチ操作で直感的な操作が可能である。指先で行う動作に

表 1 開発環境

iMac	macOS Sierra 10.12.6
ソフトウェア	Xcode 9.1
言語	Swift
タブレット端末	iPad Pro 10.5 iOS 11.1.1

表 2 タッチ操作

タップ	1 本の指で軽く叩く操作
スワイプ	一定方向へ指を動かす操作
ピンチイン/アウト	親指と人差し指をつまむ/押し広げる操作

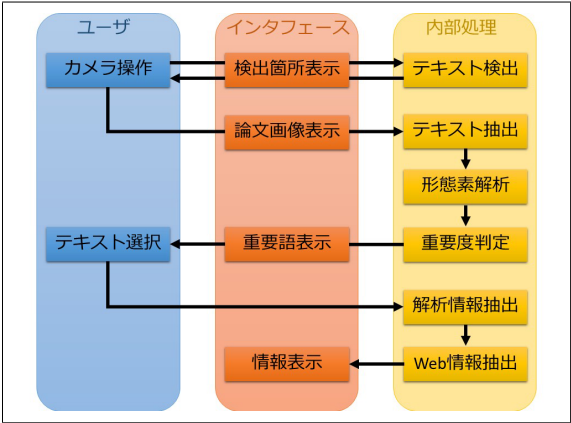


図 1 提案インタフェースにおける処理の流れ

は様々な呼称があるが、本論文では表 2 を用いる。

3.3 実装

本インタフェースにおける処理の流れを図 1 に示す。本インタフェースは起動時、まずカメラ画面が表示される。iOS のカメラ機能の実装には AVfoundation.framework^(注1) を利用している。次にカメラからの入力画像に対して、テキストをリアルタイムに検出する。ユーザは基本的に論文全体のテキストが検出されるように撮影する。次に、撮影した画像中の検出されたテキストを OCR により文字認識する。OCR は Tesseract-OCR-iOS^(注2) を利用している。つづいて、抽出したテキストを形態素解析し、単語もしくは複合語の重要度を算出する。重要度の高い重要語がインタフェース上に表示され、ユーザは重要語やテキストの検出箇所をタップすることで、興味のある語を選択できる。語を選択すると、重要度等の解析結果及び Web から収集した情報、解説ページへのリンクが表示される。

テキスト検出の実装については 3.4 節、重要語抽出については 3.5 節、また各閲覧支援機能については 4 節で詳しく述べる。

3.4 テキスト検出

本節では、本インタフェースのカメラ画像からのテキスト検出の実装について述べる。

3.4.1 Vision.framework による画像処理

Vision.framework は、iOS 11 から追加されたフレームワークで、画像処理に関する機能を提供するライブラリである [13]。このフレームワークでは、同じく iOS 11 から追加された機械

(注1) : <https://developer.apple.com/documentation/avfoundation>

(注2) : <https://github.com/gali8/Tesseract-OCR-iOS>

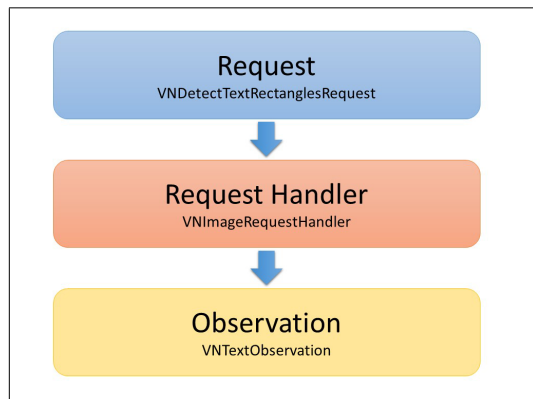


図2 Vision.frameworkによるテキスト検出の流れ

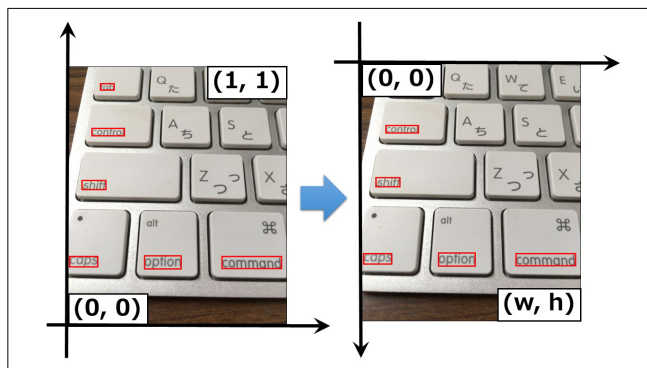


図3 座標系の変換

学習のフレームワークである Core ML が内部で使われている。

提案するインタフェースでは、Vision.framework で提供されている画像処理に関する機能の一つである、画像からのテキスト検出を行う Text Detection を利用する。Text Detection では、Apple 社により既に画像認識のモデルが用意されているため、そのまま利用することでテキストの検出を行うことができる。

Vision.framework によるテキスト検出の流れを図2に示す。図2は使用するオブジェクト名と Text Detection におけるクラス名を示している。まず、画像処理タスクをシステムに要求する Request オブジェクトを作成する。次に、検出対象の画像を保持し Request を処理するための RequestHandler オブジェクトを作成する。RequestHandler を使い Request を実行すると、処理結果として Observation オブジェクトが得られ、テキストブロックの矩形と、その各文字の矩形の座標が得られる。

3.4.2 座標変換

3.4.1項で得られる矩形座標は図3の左の画像のように、画像の左下を(0,0)、右上を(1,1)とする座標系となっている。しかし、iPadの画面の座標系は、図3の右の画像のように左上を(0,0)、右下をタブレット端末の画面の大きさとする座標系となっている。そのため、得られたテキスト矩形を描画するために、矩形座標を図3の左の座標系から右の座標系に変換する。

3.5 重要語抽出

3.5.1 複合名詞の抽出

前野らは、専門用語等の重要語の多くは名詞かつ複合名詞であるため、1単語または2単語の重要語を抽出した[2]。具体的

には、形態素解析したテキストを単語 bigram に分割し、その中で名詞もしくは未知語のみが含まれているものを抽出した。

本インタフェースでは、3単語以上に対しても解析結果が表示できるように、複合名詞の対象をN単語に変更し、以下の手順で複合名詞を抽出する。

- (1) 品詞が名詞、未知語の単語を語1として抽出する。
- (2) 語1の一つ前の単語の品詞が名詞、未知語、形容詞の場合、その単語を語2として抽出する。
- (3) 語2と語1を空白文字で連結したテキストを語1とする。
- (4) (2)の語2が得られる間(2)~(3)を繰り返す。

上記により得られた2形態素以上の最後の形態素が名詞、もしくは未知語であるものを複合名詞とする。なお、形態素解析には Swift の NSLinguisticTagger class^(注3)を用いる。NSLinguisticTagger class では各単語ごとに品詞を決定する。

3.5.2 重要度の判定

前野らは、重要語を単に頻出と言う観点からでは不十分であると判断し、TF-IDF [14] 値を算出することにより、重要語を抽出した[3]。本インタフェースも同様に TF-IDF 値により重要度を決定する。単語 t_i のその TF-IDF 値の算出式を以下に示す。

$$tfidf_i = tf_i * \log\left(\frac{num}{df_i}\right) \quad (1)$$

ここで、 tf_i はユーザが撮影した論文画像中の検出されたテキストにおける単語 t_i の出現頻度、 $num = 16,831,499$ であり、これは CiNii^(注4)における論文の総収録件数(2014年6月17日時点)、 df_i は CiNii において t_i を検索した時の検索結果数(論文数)を表す。

3.5.3 重要度算出処理の並列化

本インタフェースでは、カメラ画像から得たテキストを用い、重要語の重要度算出処理をリアルタイムに行う。ここでは、3.5.2項で述べた重要度算出処理の実行時間を短縮するために、重要度算出処理を並列で実装する方法について述べる。3.5.2項において、TF-IDF 値を算出する際に1語ずつ CiNii にアクセスし検索結果を取得すると、実行時間が長くなる。そのため、この通信処理を並列化することで実行時間を短縮する。

通信処理の並列化は、以下の二種類の API のいずれかを利用して実装する。

- URLSession^(注5)
- GCD(Grand Central Dispatch)^(注6)

URLSession は非同期で通信処理を行う API であり、複数のリクエストを同時に実行できる。また、アプリがバックグラウンドの状態になっていても通信できる。

GCD は iOS で並列処理を利用するための API である。GCD

(注3) : <https://developer.apple.com/documentation/foundation/nslinguistictag>

(注4) : <http://ci.nii.ac.jp/>

(注5) : <https://developer.apple.com/documentation/foundation/urlsession>

(注6) : <https://developer.apple.com/documentation/dispatch>

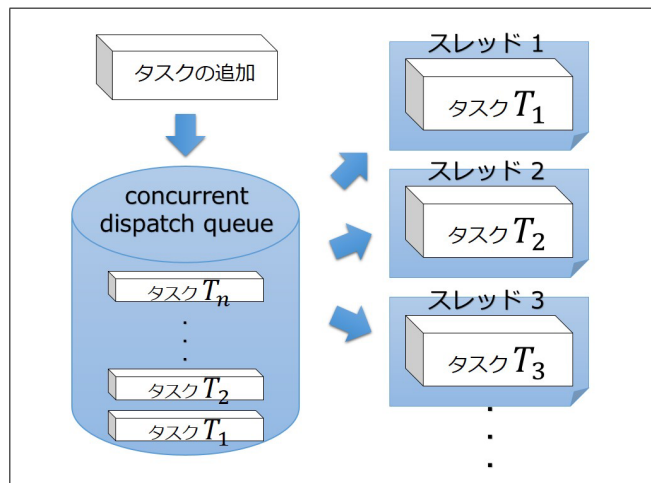


図4 GCDによる並列処理の様子

を用いてタスクを並列に行う様子を図4に示す。GCDでは、concurrent dispatch queueという並列キューに処理（タスク）を追加する。図4は、タスクを並列キューに追加し、GCDが複数のスレッドを生成してタスクを実行する様子を示している。タスクを追加すると、GCDはシステムの状態に応じてスレッドを生成し、並列に実行する。本稿では、図4のタスク $T_1 \sim T_n$ を、単語列 $t_1 \sim t_n$ のそれぞれのTF-IDF値を算出する処理と定義することで、重要度算出処理を並列化する。

4. 提案インタフェースによる学術論文閲覧支援

4.1 実行画面

実装した提案インタフェースの実行画面を図5に示す。ここでは例として、NTCIR-9のPatentMTで発表されたJeffらの論文[15]を表示している。

図5の画面にはカメラで撮影した紙媒体の論文画像が表示されている。論文画像中の検出されたテキストは赤色の矩形で囲まれている。矩形領域をタップすると、領域中のテキストがOCRにより文字認識され、画面の右側に領域中の複合名詞を含む名詞、動詞、形容詞、未知語が赤で表示される。図5では、ABSTRACTの2行目がタップされている。画面の左側には、3.5節の方法で抽出された重要度の高い上位30語が青で表示される。以後、選択した矩形領域中の語（赤）が表示されている部分をテキスト選択部、重要語（青）が表示されている部分を重要語部と呼ぶ。重要語部では、重要語の少し右を下にスワイプすることで30語までの重要語を閲覧できる。また、重要語部、テキスト選択部の語をタップすると、画面の下部に選択した語に関するWeb上の情報等が表示される。図5では、重要語部の“patent”がタップされている。以後、Web上の情報等が表示されている部分を情報提示部と呼ぶ。情報提示部でも、スワイプすることで表示情報を全て閲覧することができる。また、重要語部、テキスト選択部で選択した語は、その出現箇所が論文画像上で青く網がけされる。

右下の4つの緑色のアイコンは、それぞれタップすることでインタフェースの操作や表示の切り替えを行える。以後、上から順に1つ目のアイコンをカメラアイコン、2つ目を解析アイ

コン、3つ目を検索アイコン、4つ目を選択アイコンと呼ぶ。カメラアイコンは、カメラのオンオフの切り替えを行う。カメラをオフにして画面を止めた場合、ピンチイン、ピンチアウトにより検出テキスト画像の拡大、縮小が行える。解析アイコンは、現在画面上で検出されている全ての矩形領域のテキストを文字認識し、重要度の算出を行って重要語を表示するまでの一連の処理を行う。検索アイコンは、画面の下部の情報提示部の表示非表示を切り替える。選択アイコンは検出されている矩形領域の表示非表示を切り替える。非表示時はテキストの矩形領域をタップで選択できなくなる。

4.2 閲覧支援機能

4.2.1 重要語の表示

図5の重要語部について述べる。本インタフェースでは、表示される重要語がWikipedia^(注7)の記事に存在するかどうかで色の濃淡を変更する。青色の背景で表示されている重要語は英語のWikipediaのページに記事が存在し、濃い青色の背景で表示されている重要語は記事が存在しない。これらは、解析アイコンタップ時に予めWikipediaを検索し、記事を取得できるかどうかで判定する。なお、重要語が頭字語（例えば、MT）であり、実体語（例えば、Machine translation）が特定できない場合、Wikipediaに記事が存在するかどうか判定できない場合がある。このような場合、Wikipediaでは、頭字語の意味と一致する可能性のある語の一覧を表示することが多い。本インタフェースでは、この場合も通常の青色の背景で表示する。

また、重要語の重要度算出処理には時間がかかるため、解析開始時、まず始めに重要度を算出した10語を重要度順にインタフェース上に表示する。その後10語算出する毎にインタフェース上に表示する語を重要度順に従って更新する。全ての語の重要度を算出後、重要度の高い上位30語を重要語として表示し、それらの語がWikipediaの記事に存在するかどうかで背景の色の濃淡を変更する。

4.2.2 解析結果と関連情報の表示

図5の情報提示部の表示内容について述べる。情報提示部では、選択した語に対して、左から順に、出現頻度や重要度等の解析結果（赤）、Wikipediaの要約（橙）、Weblio^(注8)で表示される情報（緑）、Google^(注9)の上位3件の検索結果（青）を表示する。

解析結果（赤）は、検出テキスト内でのその語の出現頻度、その語が頭字語である場合には頭字語の元となる実体語、その語の初出箇所、TF-IDFに基づく重要度からなる。ここで重要度とは、その語のTF-IDFが検出テキスト内において上位から何番目であるかを表す。上位から10語を最も重要度の高い語とし、「★★★★★」と表示する。以降は10語ごとに一つの★が☆となり、51語以降は「☆☆☆☆☆」とする。

Wikipedia（橙）はWikipedia記事の要約部分の第一段落、Weblio（緑）はWeblioの新英和中辞典の情報を表示している。

(注7) : <http://en.wikipedia.org/wiki/>

(注8) : <http://ejje.weblio.jp/>

(注9) : <https://www.google.co.jp>

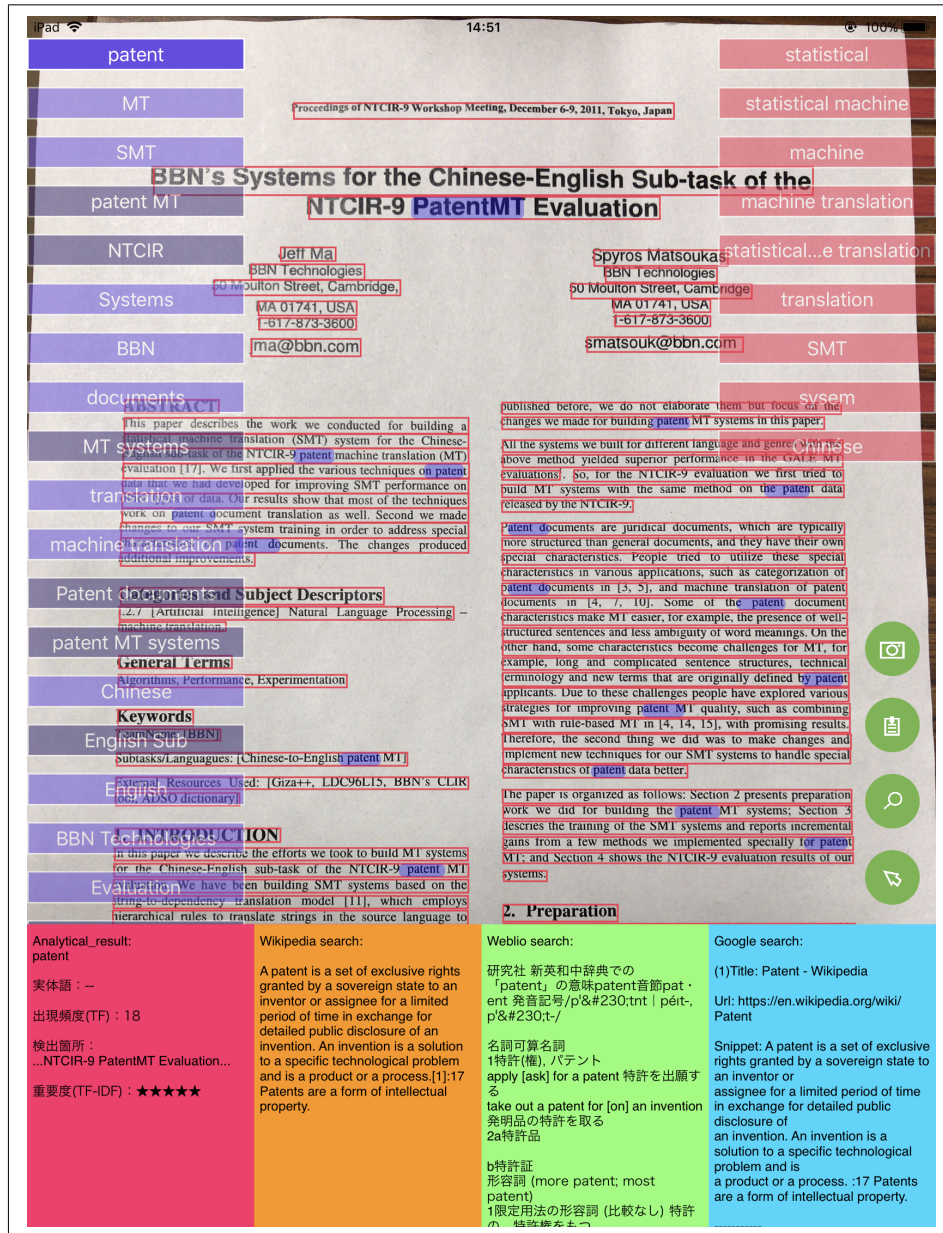


図 5 提案インタフェースの実行画面

Google (青) については、Google Custom Search API^(注10) を用いて検索結果の上位 3 件を取得し、ページタイトル、URL、スニペットを表示する。また、図 6 のように、Wikipedia (橙)、Weblio (緑)、Google (青) の情報が表示されている箇所をタップすることで、iOS のタブレット端末等に標準搭載されている Web ブラウザアプリである Safari^(注11) が起動し、それぞれの Web ページに移動できる。

4.2.3 選択した語の出現箇所の特定

図 5 に示した重要語部またはテキスト選択部の語を選択すると、その語の出現箇所が論文画像上で青く網がけされる。この機能は解析アイコンがタップされ全テキストを解析する際に、矩形領域の座標と OCR 認識結果を紐づけておくことで実装する。青く網がけする領域の左上の x 座標、 y 座標、幅、高さ

は以下の手順により取得する。

(a) ユーザが選択した語 t_i が矩形領域 $r_{[i]}$ 中のテキストに含まれている場合、その矩形領域 $r_{[i]}$ の左上の x 座標 $x_{r_{[i]}}$ 、 y 座標 $y_{r_{[i]}}$ 、幅 $w_{r_{[i]}}$ 、高さ $h_{r_{[i]}}$ を取得する。

(b) 青く網がけする領域の左上の x 座標 x_i と幅 w_i を以下の式 (2)、(3) により決定する。

$$x_i = x_{r_{[i]}} + (w_{r_{[i]}} * \frac{c_s}{c_n}) \quad (2)$$

$$w_i = w_{r_{[i]}} * \frac{c_t}{c_n} \quad (3)$$

ここで、 c_n は領域 $r_{[i]}$ 中の空白を含む全文字数、 c_s は領域 $r_{[i]}$ 中でユーザが選択した語 t_i が出現するまでに存在するテキストの文字数、 c_t は語 t_i の文字数を表す。

(c) 青く網がけする領域の左上の y 座標と高さをそれぞれ $y_{r_{[i]}}$ 、 $h_{r_{[i]}}$ とする。

これらを全てのテキストの矩形領域について行う。

(注10) : <https://developers.google.com/custom-search/?hl=ja>

(注11) : <https://www.apple.com/jp/safari/>

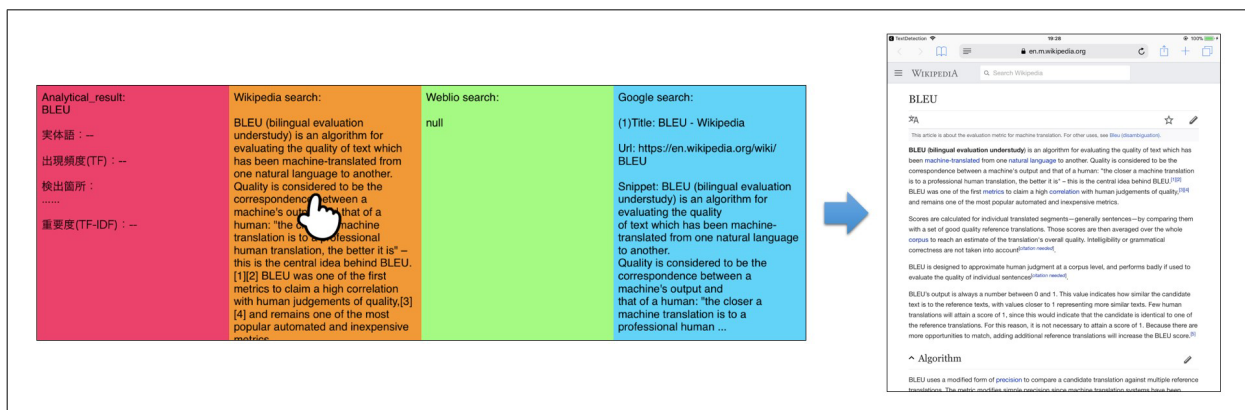


図 6 Wikipedia(橙) を選択した場合のリンク先の様子

5. 評価実験

2018 年 1 月に、岡山大学工学部情報系学科の 4 年生 3 名と
同大学院自然科学研究科の博士前期課程の 1 年生 3 名と 2 年生
3 名の計 9 名に Jeff らの論文 [15] を読み、論文中の指定された
語の意味を調べるという課題に取り組ませた。

5.1 実験概要

学術論文等の専門性の高い文書をタブレット端末で読む場合、
専門用語の意味を検索したり、他ページに戻って内容を確認し
たりするなど、ページ間、また複数文書やメディア間での移動
が発生する。もしこのようなインタラクションを減らすことが
できれば、論文理解の時間短縮になる。そこで本節では、被験
者に提案インタフェースを用いて課題に取り組ませ、その使用
時間とインタラクション数を計測し評価する。

本実験では、提案インタフェースの比較対象として前野の
マルチインタフェース [1]、iPad の電子書籍閲覧用アプリケー
ションの一つである iBooks を用いる。なお、これらのインタ
フェースでは論文 PDF を用いるが、提案インタフェースでは
印刷した紙媒体の論文を用いる。被験者は、各インタフェース
を用いて設定した課題に取り組む、また課題終了後に提案イン
タフェースに関するアンケートに答える。

課題は、被験者に閲覧させる論文中に表れる語の意味を調べ
させるものである。まず被験者に各インタフェースの使い方を
教え、課題の取り組み方について説明する。その後、各インタ
フェースを使用し、文中に表れる語に対して、どのような意味
であるのか調べさせる。調べさせる語は以下の 9 つである。

- (1) corpus
- (2) SMT
- (3) gradient descent
- (4) juridical
- (5) BLEU
- (6) dependency tree
- (7) newswire
- (8) CLIR
- (9) word alignment

被験者にはまず指定のインタフェースを用いて (1)～(3) の意
味を調べさせ、インタフェースを変更して (4)～(6)、残りのイ

ンタフェースで (7)～(9) を調べさせる。インタフェースを使用
する順番は被験者毎に変更し、各語に対する各インタフェース
の被験者全体の合計使用回数が同じとなるようにする。選択し
た 9 問の課題は以下の 3 種類に分類できる。

- 一般的な単語 (1, 4, 7)
- 専門性が高いと思われる頭字語 (2, 5, 8)
- 語の意味等の解説が必要と思われる複合語 (3, 6, 9)

実験では、被験者が各インタフェースを使用している間の
iPad の画面を録画し、各インタフェースでの課題を終えるまで
の経過時間、また語の説明が表記されている解説ページに辿り
着くまでのインタラクション数を計測する。また、本実験で定
めた各インタフェースにおけるインタラクションを表 3 にまと
める。本実験では、インタラクションは語の選択時と解答の探
索時に発生すると定め、それぞれのインタラクション数を全て
1 として数える。また、Web ブラウザ等の別のアプリケーシ
ョンへの画面遷移と、Web ブラウザ内でのページ移動が発生した
場合もインタラクション数を 1 として数える。例えば、提案イ
ンタフェースを用いて語を以下のように調べた場合、表 3 に定
める合計インタラクション数は 5 となる。

- (1) テキスト選択部から語を選択 (b)
- (2) 情報提示部で解答を探す (c)
- (3) Google の検索結果へ移動 (m) して解答を探す (d)
- (4) 該当と思われるページへ移動して解答を探す (l)

提案インタフェースでは、重要語部とテキスト選択部から語
を選択できる。重要語部は解析アイコンをタップすると自動
で表示される。テキスト選択部は検出されている矩形領域を
タップすると表示される。マルチインタフェースでは、“Select
Text” というボタンを押した後、語の左上と右下をタップす
ることで 1 単語または 2 単語のテキストを選択できる。その後表
示される情報提示窓の表示内容は提案インタフェースの情報提
示部と同じである。iBooks では、テキストを長押しすると長押
しした部分に最も近い単語が青く網がけされる。その後選択範
囲を変更することができ、ユーザは自由に語を選択できる。語
を選択後、日本語の Wikipedia のページと Google の検索結果
のページへ移動できる。

被験者が各インタフェースを用いても解答を見つけられな
かった場合には、サーチエンジンを用いて意味を探させる。そ

表 3 各インタフェースにおけるインタラクション

提案インタフェース	a. 重要語部から語を選択
	b. テキスト選択部から語を選択
	c. 情報提示部で解答を探す
	d. リンク先で解答を探す
マルチインタフェース	e. “Select Text” で語を選択
	f. 情報提示窓で解答を探す
	g. リンク先で解答を探す
iBooks	h. 長押しで語を選択
	i. “Wikipedia で検索” で解答を探す
	j. “WEB を検索” で解答を探す
共通	k. クエリを変更する
	l. ページを移動して解答を探す
	m. 画面遷移が発生する

の際、サーチエンジンにクエリを入力した場合もインタラクション数を 1 として数える。調べる順番は特に限定せず、被験者には自由に語の意味を調べさせる。また、被験者が元々語の意味を知っていた場合も、各インタフェースを用いて語の説明が書かれている解説ページを探させ、時間とインタラクション数を計測する。

5.2 実験結果

課題の語を調べるのににかかった時間とインタラクション数をそれぞれ図 7 と図 8、被験者毎の各インタフェースの使用時間を図 9、提案インタフェースにおける操作毎の使用時間を図 10 に示す。

図 7、図 8 の (1)～(3) は、各インタフェースを用いて調べさせた語の課題番号を示している。図 7 を見ると、全体的に提案インタフェースが時間がかかっており、iBooks が最も短いという結果となった。各被験者の使用動画を見ると、全体的にインタフェースの動作は iBooks が最も軽く、マルチインタフェースや提案インタフェースは操作中に起動停止した場面があった。よって、インタフェースの処理の重さや予期していない動作時のエラー処理に対応できていない点がこの原因の一つと考えられる。

図 8 を見ると、全体の平均では提案インタフェースにおけるインタラクション数が最も少なく、iBooks が最も多いという結果となった。特に (1) の “corpus” や (4) の “juridical” のような一般的な語は、提案インタフェースの情報提示部やマルチインタフェースの情報提示窓の Weblio の検索結果に語の意味が提示されたため、画面遷移によるインタラクションが少なかった。しかし、マルチインタフェースの場合、“Select Text” で語を選択するときに操作誤りが発生し、語の選択をやり直していたため、提案インタフェースと比較するとインタラクション数が多くなった。また、iBooks の場合は語の意味を調べる際に多くの画面遷移が発生したため、インタラクション数が最も多くなった。iBooks から Wikipedia ページに移動したが、Wikipedia 記事が無かったため iBooks に戻って Google ページに移動するパターンが多く、提案インタフェースでは情報提示部で予め Wikipedia の記事が存在するかどうか分かる点が有効であった。

図 9 の被験者毎の結果を見ると、被験者 5、6、9 では提案インタフェースの使用時間が他のインタフェースと比べて極端に長くなっていた。被験者 5、6、9 の提案インタフェースの使用動画を確認すると、被験者が撮影した紙論文の画像に対して検出したテキスト領域が傾いていたり、検出箇所の OCR による文字認識が誤っていたりしたため、紙論文の撮影のやり直しを 1 回以上行っていた。特に、被験者 6 と被験者 9 は 3 回以上の撮影のやり直しが発生していたため、テキスト領域の傾き補正や文字認識誤りの訂正処理を検討する必要がある。

図 10 では、提案インタフェースにおける撮影にかかった時間、解析アイコンをタップして重要語を表示するまでにかかった時間、実際に語を調べた時間を示している。図 10 を見ると、被験者 5、6、9 は撮影のやり直しが何度も発生したため、撮影時間と解析時間に時間がかかっている。撮影時間と解析時間を含めなければ、被験者 1 と被験者 3 では図 9 の iBooks の使用時間よりも提案インタフェースの方が短い結果となる。これは、被験者 1 と被験者 3 が提案インタフェースで調べた課題の語のうちの一つが重要語部に表示されたため、語の選択が容易であったことや、情報提示部で語の意味を得られた点が有効であったからである。

課題終了後、被験者に提案インタフェースと他のインタフェースとの差異等について意見を自由に記述させた。それによると、提案インタフェースは語の選択が楽と答えた被験者が 9 名中 5 名いた。提案インタフェースではタップのみで調べたい語が右のテキスト選択部に表示されるため、iBooks やマルチインタフェースのように語の領域を被験者自身が指定する必要がなかった。また、調べたい語が含まれているテキスト領域をタップすることで、その領域の単語と 2 単語以上の複合名詞がテキスト選択部に表示される。例えば、“dependency” と “dependency trees” の両方が表示される。この点が便利であったという意見も得られた。また、情報提示部の情報がマルチインタフェースと比べて見やすいと答えた被験者が 2 名いた。マルチインタフェースの情報提示窓は、解析結果や検索結果をスワイプして閲覧する必要があったが、提案インタフェースでは全ての検索結果を同時に閲覧できるため、役に立ちそうな情報を見つけやすく、そのリンク先のみに移動することで効率的に検索できるという意見が得られた。また、最初に紙論文を撮影するカメラ操作が難しいと答えた被験者が 4 名いた。カメラ操作にブレが生じてテキストの検出領域が傾いてしまったり、テキスト領域中の文字認識誤りが発生したりして何度もやり直すことも多かった。そのため傾き検出等の画像処理の追加や、文字認識誤りへの対応が今後の課題と考える。

6. ま と め

本稿では、iPad のカメラ機能を用いて紙媒体の論文画像を取得し、画像中のテキストに対して解析結果や Web 情報を提示するインタフェースを提案した。被験者実験では、提案インタフェース、前野のマルチインタフェース、iBooks で課題として設定した語の意味を調べ終えるまでの時間、解説ページに到達するまでのインタラクション数を計測し比較した。その結

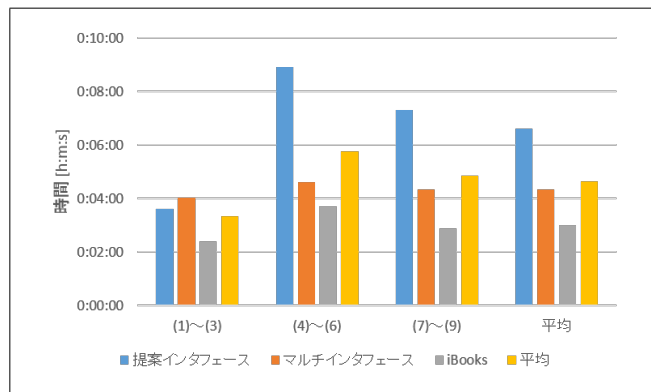


図 7 課題にかかった時間

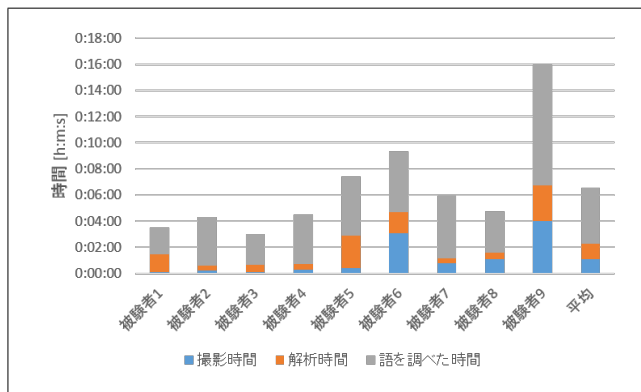


図 10 提案インタフェースにおける操作毎の使用時間

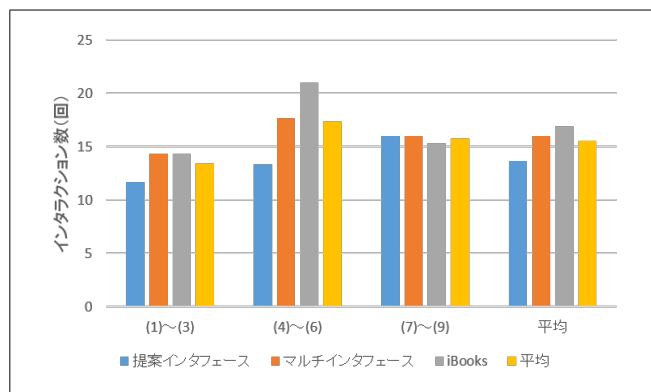


図 8 課題にかかったインタラクション数

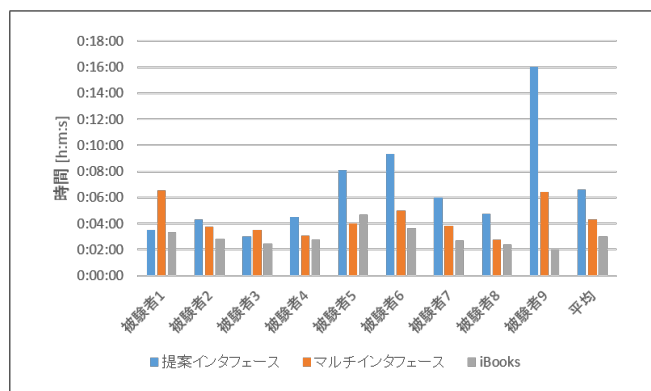


図 9 被験者毎の各インタフェースの使用時間

果、使用時間は iBooks が最も短い結果となったが、インタラクション数は提案インタフェースが最も少ない結果となった。

今後の課題としては、画像の傾き検出処理や、文字認識誤りの訂正処理の検討の他に、メモ機能等の閲覧支援機能の拡充が挙げられる。提案インタフェースでは画像であれば様々なテキストに対して閲覧支援が可能であるため、例えば学術論文のポスターなどにも対応できれば有用と考えられる。

謝 辞

本研究の一部は、国立情報学研究所公募型共同研究の援助による。ここに記して深謝する。

文 献

- [1] 前野明子, “電子書籍閲覧端末による学術論文閲覧支援インタフェースに関する研究”, 岡山大学大学院自然科学研究科修士論文, 2015.
- [2] 前野明子, 太田学, 高須淳宏, “学術論文閲覧支援インタフェースの試作”, 第 6 回データ工学と情報マネジメントに関するフォーラム (DEIM 2014), E3-3, 2014.
- [3] 前野明子, 太田学, 高須淳宏, “学術論文閲覧支援インタフェースのための頭字語の活用”, 第 160 回データベースシステム研究会, Vol. 2014-DBS-160, No. 16, pp. 1-8, 2014.
- [4] 柴田 博仁, 高野 健太郎, 大村 賢悟, “電子書籍端末は紙を代替できるか? 電子書籍端末の評価実験にもとづく考察”, 富士ゼロックス テクニカルレポート, No. 21, 2012.
- [5] 小林亮太, 池内淳, “表示媒体が文章理解と記憶に及ぼす影響—電子書籍端末と紙媒体の比較—”, 研究報告ヒューマンコンピュータインタラクション (HCI), pp. 1-7, 2012.
- [6] 阿辺川武, 相澤彰子, “内部構造解析機能と脚注表示機能を備えた論文閲覧システム”, 人工知能学会インタラクティブ, 情報アクセスと可視化マイニング第 7 回研究会, pp. 13-18, 2014.
- [7] 川上優平, 清水敏之, 吉川正俊, “論文とプレゼンテーションスライドの部分対応付けにおける軸対応補正の適用”, 第 77 回全国大会講演論文集, pp. 795-796, 2015.
- [8] Yuhei Kawakami, Atsuto Nishida, Toshiyuki Shimizu, Masatoshi Yoshikawa, “Axis-based Alignment of Scholarly Papers and Its Presentation Slides Considering Document Structure”, ICADL, LNCS 8839, pp. 87-97, 2014.
- [9] 鉢木稔浩, 太田学, 高須淳宏, “Web 資源を利用した学術論文閲覧支援システム”, 情報処理学会研究報告, Vol. 2009-DBS-149, No. 14, pp. 1-6, 2009.
- [10] 鉢木稔浩, 太田学, 高須淳宏, “学術論文閲覧支援システムのための関連論文推薦”, 第 3 回データ工学と情報マネジメントに関するフォーラム (DEIM 2011), F9-4, 2011.
- [11] 今井智宏, 望月久稔, “共起関係と係り受け関係を導入した文書グラフの解析による特徴ベクトルの抽出”, 第 7 回データ工学と情報マネジメントに関するフォーラム (DEIM 2015), A2-3, 2015.
- [12] 小倉由佳里, 小林一郎, “PageRank アルゴリズムを用いた重要文抽出による潜在意味に基づく文書分類への取り組み”, 第 19 回言語処理学会年次大会, pp. 690-693, 2013.
- [13] Vision | Apple Developer Documentation, <https://developer.apple.com/documentation/vision>
- [14] Gerard Salton, Edward A. Fox, Harry Wu, “Extended Boolean Information Retrieval”, Communications of the ACM, pp. 1022-1036, 1983.
- [15] Jeff Ma, Spyros Matshoukas, “BBN’s Systems for the Chinese-English Sub-task of the NTCIR-9 PatentMT Evaluation”, Proceedings of NTCIR-9 Workshop Meeting, pp. 579-584, 2011.