【ノート】

全文検索による試験問題検索システム

― 新規作成問題の類似文書検索を中心として―

石岡恒憲（大学入試センター），狩野芳伸（静岡大学）
橋本貴充，大津起夫（大学入試センター）

全文検索で有名な Hyper Estraier を大学入試センター研究開発部門のネットワーク上にある Linux サーバにインストールし，特定のディレクトリに格納されているセンター試験問題 PDF を Web ブラウザから検索する環境を整えた。ユーザは Web 上の特定のアドレスを指定することでキーワード検索ができる。また，ユーザは新作した試験問題について，過去に作成された類似度の高い文書を網羅的に検索することができる。これらのシステムはソースコードから構築しており，ユーザインターフェイスの改良だけでなく，今後のシステム拡張を容易にしている。

1　はじめに

大学入試センター試験では多大な注意をはらって試験問題が作成されている。注意の種類はさまざまであり，試験問題としての適切性を判断する点検作業には多くの困難が伴う。その困難の一つは作成および点検すべき教科・科目の数が多いことによる。センター試験において地理歴史と公民は，平成 24 年度より時間割の区別がなくなり，「世界史Ａ」「世界史Ｂ」「日本史Ａ」「日本史Ｂ」「地理Ａ」「地理Ｂ」「現代社会」「倫理」「政治・経済」「倫理，政治・経済」の 10 科目（グループ）から最大 2 科目を選択解答することになった。また，平成 27 年度からは，例えば理科において，理科①「物理基礎」「化学基礎」「生物基礎」「地学基礎」の４科目，理科②「物理」「化学」「生物」「地学」の４科目に加え，理科②の経過措置として「理科総合Ａ」「理科総合Ｂ」「物理Ⅰ」「化学Ⅰ」「生物Ⅰ」「地学Ⅰ」の６科目が実施される予定である。

このような多くの教科・科目が同時に実施される場合に，同時に実施される科目間の重複についても，また過去の試験についても，さらに比較的近年の有名国公私立大学の入試問題についても調査し，試験問題の重複や類似性を検討することは，もはや人手でできるものではなく，コンピュータによる支援は不可欠である。

キーワードを指定して，PDF を含むコンピュータ内の文書ファイルに含まれる文字列を検索するには，かつては Google のデスクトップ検索（2011 年 9 月にサービスを終了）があり，現在では Windows の「プログラムとファイルの検索」機能がある。また，全文検索システムとして Hyper Estraier や Namazu 等，多くが存在するが，本稿はこれらシステムを比較・評価するものではない。我々は，ある新作の問題を電子媒体上で作った場合に，それが既に作られている問題と内容の，少なくとも文面上のキーワードに類似性がないかを判断し，類似性が高いと推測される文書を「（洩れなく）網羅的に」検索するシステムを要求する。またそれ以外に以下の要件を満たすことが必要となる：

- 縦書きの文書についても検索可能なこと（センター試験では国語を含むため）。
- PDF や Word，Excel で書かれたファイルの中身についてもキーワード検索できること（センター試験の過去問は PDF で作成の上，コンピュータ上に格納され

ている。また作題者は多くの場合，Word，Excel で試験問題を作成する。）

- 試験問題では用語が統一されており，通常，検索システムで要求される，意味的な文書の一致や表記の揺らぎの吸収についてはさほど重要でない。それよりも，検索による洩れのないことを最優先する。
- ある程度のシステム応答性，つまり使っていて違和感のない程度には検索結果が返ってくること。
- システム検索の仕組みがブラックボックスではなく，その仕様が公開されていること。また，必要に応じてシステム改良ができるように，ソースコードが公開されており，コマンドベースで起動できること。

以上を勘案し，我々は全文検索システムとして Hyper Estraier（平林, 2007）を使用し，新規作成問題の類似文書検索を Web ベースで検索するシステムを，大学入試センター研究開発部内のサーバ上に構築した。

本稿では，検索システムについて，その仕様を示すとともに，今後の方向性や果たすべき役割について論考する。2 節では，動作させるための前提や，必要とするシステム要件について整理しておく。3 節では，ユーザからみたシステムの操作手順や見た目についてスクリーンダンプを示しながら説明する。4 節には，システムを保守してゆくための必要な手順について説明する。5 節はキーワード抜き出しの性能を向上させるためのユーザ辞書の設定について述べる。6 節には，まとめと今後について述べる。本稿は同様のシステムを構築しやすいよう，あえてシステム構成（2 節），保守（4 節），辞書の設定の手順（5 節）について詳細に説明し，その部分についてはリファレンスマニュアル風になっていることを断っておく。

2　システム要件

Hyper Estraier（平林，2007）はインデックスを使った高速な全文検索システムの一つであり，形態素解析と N-gram のハイブリッド機構で検索精度を向上させている。Linux2.2 以降等で動作する。

2.1　事前準備

Hyper Estraierをインストールするに際し，前もって以下の3つのライブラリを入れておく必要がある。

（1）libiconv：GNUプロジェクトによる文字コード変換ライブラリ。バージョン1.9.1以降が必要。これはGNUプロジェクトにおけるCの標準ライブラリglibcにも同梱されている。インストールは簡単で，ソースコードから`./configure; make; make install`でできる。

（2）zlib：データの圧縮および伸張を行うためのフリーのライブラリ。可逆圧縮アルゴリズムのDeflate (RFC 1951)を実装している。バージョン1.2.1以降。Linuxディストリビューションの一つであるCentOSでは，予めインストールされている。

（3）QDBM： 組み込み用データベースを扱うルーチン群のライブラリ。バージョン1.8.75以降。これはソースコードから
`./configure; make; make install`
でインストールできる。

2.2　コンフィグレーション

標準の環境では（特に指定しなければ），検索のインデックスは，Web コンテンツの格納ディレクトリ（たとえば/var/www/html/）以下の casket に置かれる。Hyper Estraier のコンフィグレーション・ファイルは標準では
/var/www/cgi-bin/estseek.conf
に置かれ，このファイルの中身のうち `indexname` と `replace` だけは変更する必要がある。たとえば，以下のように書きかえる。

```
indexname: /var/www/html/casket
replace: file:///var/www/html/{{!}}
  http://ae01.rd.dnc.ac.jp/
```

ここで `indexname` にはインデックスのパス

を指定する。replace はローカル用の URI を Web サーバ用の URI に変換する指定であり，「{{!}}」の前後にローカルの接頭辞と Web サーバ用の接頭辞を書く。

2.3 カスタマイズ

（1）標準の設定では PDF ファイルが索引付されないため，以下の通り対処する。

PDF ファイルも検索の対象に含めるにはコマンド estfxpdftohtml が必要で，このコマンドは「hyperestraier の展開先/filter/」に格納されてある。Web では実行コマンドは nobady のユーザ設定で実行されるので，root の PATH を通すことをせずに，/usr/local/bin にこのコマンドファイルをコピーする。

（2）索引を作成する。このために以下のコマンドを root 権限で実行する。

```
# estcmd gather -il ja -sd -fx .pdf
 T@"pdftotext -raw"
 /var/www/html/casket
 /var/www/html/DNC;
```

これにより/var/www/html/DNC 以下の文書を収集して，casket という名前のインデックスを作る。「-il ja」オプションは，文字コードの判定時に日本語を優先するという意味である；「-sd」オプションは，ファイルの作成日時と更新日時をインデックスに記録するという意味である；「T@pdftotext」は外部コマンド pdftotext の出力がテキストなので，テキストとして解釈するという意味である；-raw なるオプションは「ルビ付きされている漢字文字列」をキーワードとして検索するための対処である。-metahtml オプションにより検索結果にタイトルや著者が表示される。Word 文書のタイトルは中身と合致しない場合があり，また文書の作成者名が表示されることは試験問題検索においては好ましくないので，通常は有益であるこのオプションはあえて付けない。

コマンドを実行すると，進捗状態のログメッセージが出てから，casket というディレクトリが作成される。インデックス生成は文書のサイズやタイプにも依るが，30 年分のセンター試験問題全科目で 1.5 時間程度を要した。

3 使い方

今回，われわれがユーザに対して開放したユーザインターフェイスは 2 つある。一つはキーワードを入力し，関連する試験問題を検索するものである。Google のような検索インターフェイスを連想してかまわない。ヒットしたファイルの内容が数行でスニペット表示され，該当ファイルをクリックするとその中身が別のウィンドウで表示される。もう一つは，ローカルのコンピュータ上にある新作の試験問題ファイルを指定すれば，そのファイルに含まれているキーワードをシステムが自動的に抽出して，そのキーワードごとに関連する試験問題を検索・表示するものである。以下，順に説明する。

3.1 キーワード検索

検索は Web ブラウザで，以下のアドレスを入力することにより起動する。

http://ae01.rd.dnc.ac.jp/cgi-bin/estseek.cgi

図 1 は，検索窓に「小選挙区比例代表並立制」を入力して「Search」ボタンを押下して得られた結果である。27 件がヒットし，2011 年「現代社会」追試問題第 2 問(2011-C1-T-02-z03.pdf)や 2013 年「倫理・政治経済」追試問題第 4 問(2013-C4-T-04-z02.pdf)が先頭に表示されている。「現代社会」と「倫理・政治経済」は試験問題の内容に少しの重複部分があることがわかる。

図 2 は，平成 25 年度の国語入試問題で話題になった「小林秀雄」を検索窓に入力して得られた結果である。平成 18 年(2006 年）の追試試験にも関連する問題が出されているほか，平成 19 年（2007 年）の倫理でも題材として取り上げられていることがわかる。また，

図１：キーワード検索（「小選挙区比例代表並立制」と入力）

国語の試験問題(PDF)は縦書きで作成されているが，この場合でも正しくキーワード検索されていることがわかる。

図２：キーワード検索（「小林秀雄」と入力）

キーワード検索を行えば，意外に多くの科目間にわたって，すなわち複数の科目で１つのキーワードでヒットすることがわかる。紙面の都合でスクリーンダンプを載せることはしないが，たとえば「マックス・ウェーバー」を入力すると，昭和 58 年（1983 年）の「倫理」と昭和 55 年（1980 年）「国語」のそれぞれの試行テスト問題でヒットする。

キーワード検索では，もちろん複数のキーワードを入れることができ，それらの論理条件（たとえば AND 条件や OR 条件）を指定することができる。Google などの検索エンジンでは，空白で区切ることで AND 条件を指定する（一般にいうところの「簡便書式」を使っている）が，Hyper Estraier でもそのような挙動の方がユーザには馴染み深いと考え，それを標準の設定とする。簡便書式では空白または「&」で区切ると AND 条件となる。複数の検索語を「""」で括るとフレーズ検索となる。「!」で NOT 検索，「|」で OR 検索である。「|」の優先順位は「&」や「!」よりも高い（「通常書式」と同じ）。また「*」によるワイルドカード検索ができ，たとえば「euro*」で「euro」で始まる単語を含む文書検索ができる。

詳しい使い方はトップページの「help」に記載してある。また「Advanced」でより高度な検索ができ，作成した文書の属性（作成日，作成者など）による検索ができる（図３）。

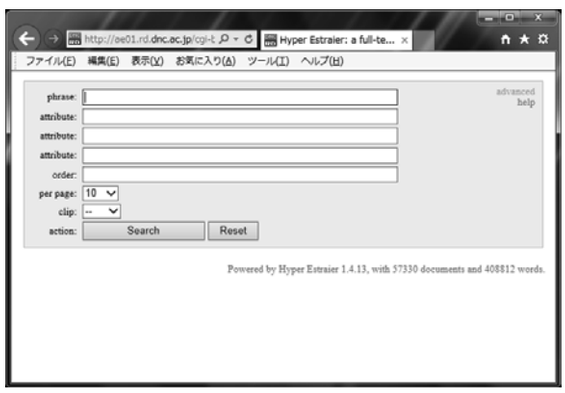

図３：より高度な検索

3.2 関連文書検索

Web ブラウザで，以下のアドレスを入力する。

http://ae01.rd.dnc.ac.jp/dncestfile/

関連文書検索では，キーワード検索とは異なり，入力は検索語ではなく，自然言語で書かれたファイルである。ここでは新規に作成した試験問題を想定している。このため，ファイルの拡張子としては `.pdf` と `.doc[x]` と `.txt` の 3 種類とする（これ以外のファイルタイプにはエラーメッセージを出力する）。

図４はこの関連文書検索の初期画面である。入力ファイルは，通常，クライアント側にあるローカルなマシンにあると思われるので，

「参照」により当該ファイルを選択指定する。

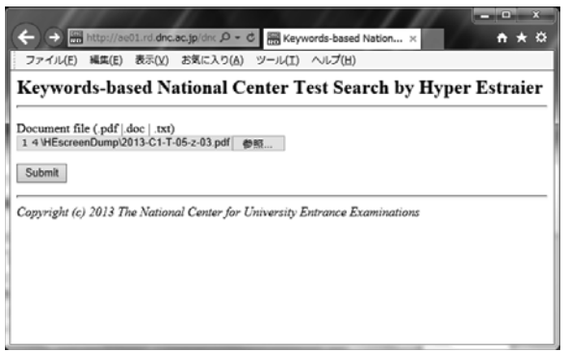

図４：関連文書検索の初期画面

実際の検索コマンドの実体は cgi-bin/relatedDocRet.pl なる自作の 200 行程度の perl プログラムである。これを動作させるために，perl のライブラリである cgi-lib.pl が別途，必要である。これを同じディレクトリに置く。また形態素解析システムとして mecab(工藤，2006)が必要である。自作 perl プログラムの動作の流れは以下の通りである：

1. 入力したファイルのテキスト化を行う。すなわち PDF ファイルに対しては，`pdftotext` コマンドを`-raw` オプション付きで，Word ファイルについては `wvWare` を実行する。テキストファイルについては `nkf` を実行し，それぞれ `utf-8` の漢字コードで格納する。

2. 行末の改行コードを削除した上で，ファイルに含まれるキーワードを抽出する。ファイルに形態素解析である mecab をかけ，それに自作の awk プログラムを掛けることでキーワードを抽出する。mecab のバッファは標準の 10 倍程度に設定する。キーワードとして固有名詞と一般名詞を抽出するようにしたが，その後，Wikipedia 見出し語をユーザ辞書と登録し，これを利用できるようにした。

3. 抽出したキーワードをソートし，重複をなくし，キーワードリストを作成し，それぞれのキーワードに対して，`estcmd search` を`-vh` オプション付きで実行する。`-vh` は属性情報とスニペットを読みやすい形で表示するための指定である。

図５は，図４で指定した検索画面からの検索結果である。入力ファイルは，小選挙区比例代表並立制についての記述のある平成 25 年度の「現代社会」の試験問題である。このファイルから複数のキーワードが自動的に抽出される。ここでは，その最初のキーワードとして「ねじれ国会」が選ばれ，「ねじれ国会」に関連する試験問題して 2011 年の「政治・経済」の問題がヒットしている。

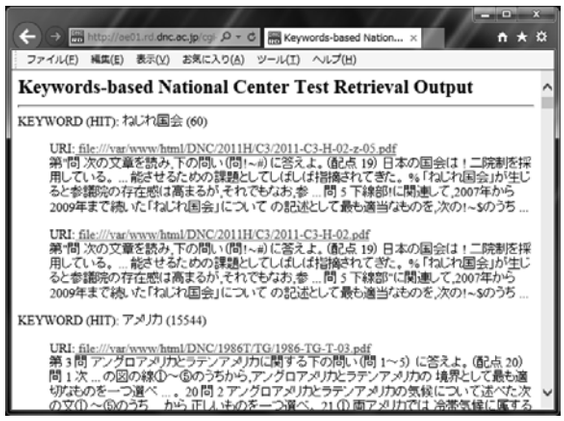

図５：関連文書検索の検索結果（「ねじれ国会」がキーワードとして抽出されている）

ここで注意したいのは，(一般的な用語である)「ねじれ」や「国会」がキーワードとして選ばれることはなく，「ねじれ国会」がキーワードとして選ばれているという事象である。これは 5 節で述べているように，ユーザ辞書として Wikipedia の見出し語を用いているためである。もし「ねじれ」がキーワードとして選択されれば，「地学」での「物質のずれやねじれ」に関する問題もヒットすることになったであろう。

なお参考までに，この例で自動的に検出されたキーワードを全て列挙すれば以下のようになる：ねじれ国会，アメリカ，マニフェス

ト，安全保障政策，公職選挙法，国政選挙，参議院，参議院議員選挙，衆議院，衆議院の優越，重複立候補制度，小選挙区，小選挙区比例代表並立制，選挙運動，選挙制度，多数派，二院制，日本の選挙，比例代表制，非拘束名簿式，有権者。これからも妥当と思える字句が選択されている様子が見てわかる。また「衆議院の優越」や「日本の選挙」のように「の」で繋ぐことで始めてキーワードになるような字句も適切に選択されていることがわかる。

3.3　国語や社会以外の学科目における検索

本システムでは全ての教科科目に対して，同一の手順で検索のインデックスを作成し，検索結果を返す。このため，検索キーワードとして「三角関数」を入力すれば，試験問題文に「三角関数」という文字列を含むファイルがヒットする。数式の一部である「sin」を入力すれば，数学の試験問題である PDF から「sin」なる文字列が切り出されるので，結果として「sin」なる文字列を含むファイルがヒットする（図６）。LaTeX 形式による数式の検索はサポートしていない。

図６：「sin」を入力とするキーワード検索

「発音」と入力すれば，試験問題文に「発音」なる文字列が含まれる（多くの場合は語学の）試験問題が網羅的に検索されることになる。英文字の検索については語尾変化を考慮しないため，「euro*」などのワイルドカード検索や OR 条件の指定等で対応する。

4　システムの保守管理

システムを運用するために以下の作業が必要である。

（1）インデックスの更新：検索対象となる文書が追加されるたび，以下のコマンドを root 権限で実行する。

```
# cd /var/www/html
# estcmd gather -cl -il ja -sd -cm casket
  /var/www/html/DNC
```

最初に「-sd」オプションをつけてファイルの更新時刻を記録していたのは，差分登録のためである。更新時刻を記録したインデックスに対しては，「-cl」と「-sd」と「-cm」オプションをつけて更新を行うことで，差分登録を行うことができる。

（2）削除文書の反映：サイト内の文書が削除された場合も，それをインデックスに反映して検索にヒットしないようにする。

```
# cd /var/www/html
# estcmd purge -cl casket
```

（3）更新作業の自動化：コマンドを定期的に実行する cron の機能を使ってインデックスの更新作業を自動化する。具体的には /etc/cron.daily/estcmd.cron なるファイルを作成し，毎日定時に実行されるべくこのディレクトリに置く。root 権限で# chmod 755 し，実行ビットを立てる。ファイルの中身は以下の通りである。

```
#!/bin/sh
/usr/local/bin/estcmd gather -cl -il ja
  -sd -cm -fx .pdf T@"pdftotext -raw"
  /var/www/html/casket
  /var/www/html/DNC;
/usr/local/bin/estcmd purge -cl
  /var/www/html/casket;
```

5　キーワード検出のためのユーザ辞書登録

関連文書検索において入力ファイルからキ

ーワードを自動抽出する際，辞書としてmecab の標準辞書である IPA 辞書を用いると，一般的な用語が選択されやすい。一方，国立情報学研究所が中心になって進めている「ロボットは東大に入れるか」プロジェクトにおいて，現在センター試験社会科の自動解答で最もよい成績を出している研究から，キーワード検出にはWikipediaの見出し語が有効であることがわかってきた(狩野，2014)。これより本システムでは通常のIPA辞書に加え，140 万語(異表記表現を含む)の Wikipedia 見出し語を mecab のユーザ辞書として登録した。この Wikipedia 見出し語を IPA 辞書にある用語よりも優先させるために，mecab の連結重みにマイナスの値を与え，また重複文字列のある複数単語においては長い単語が選ばれるよう工夫した。たとえば「東京」と「東京大学」のいずれもが登録されている状況において，東京大学が入力された場合には「東京大学」のみが出力されるよう，重みの値を文字列長さ(length)の関数とした。より具体的には，mecab の重みは 2 バイト整数，すなわち－32868 から 32867 までであるため，重みを

max(－32868, (int)(－400 * length**1.5))

とした。19 文字以上で－32868 となる。

このようして整理したユーザ辞書 wikipedia.csv を mecab のユーザ辞書として登録すべく以下のコマンドを実行した。

```
$ /usr/local/libexec/mecab/mecab-dict-index -d
/usr/local/lib/mecab/dic/ipadic/ -u
wikipedia.dic -f utf8 -t utf8 wikipedia.csv
```

-d DIR: システム辞書があるディレクトリ
-u FILE: FILE というユーザファイルを作成
-f charset: CSV の文字コード
-t charset: バイナリ辞書の文字コード

これにより，カレントのディレクトリに wikipedia.dic ができる。これをたとえば /var/www/html/wikipedia.dic にコピーし，/usr/local/etc/mecabrc に以下を追加する。

```
userdic = /var/www/html/wikipedia.dic
```

6　おわりに

全文検索システム Hyper Estraier はコマンドベースで起動することができるので，自作のスクリプトでその動作を容易に制御できる。また，ユーザインターフェイスを含むシステム改良も容易である。そのノウハウを本稿では示したつもりである。

システム構築にあたり，肝心の検索エンジンがブラックボックスでなく，その中身がわかっていることは心強い。さらなる改良を行う用意があるので，利用者のフィードバックを期待している。

なお自作した perl プログラム一式は，以下の URL にて公開している。

http://coca.rd.dnc.ac.jp/HyperEstraier/

一部の perl プログラムは自作の awk プログラムを呼んでおり，これもまた併せて同梱している。ただし，wikipedia 辞書や検索のためのインデックスはこのマシン上には構築しておらず，このままでは動作しない。また環境を構築するための Makefile も用意していない。あくまでもソースコードを公開するのみである。検索の対象となる試験問題については，株式会社ジェイシー教育研究所が販売する「センターTen」を購入すれば，過去のセンター試験のみならず多くの国公私大の試験問題を入手することができる。

参考文献

平林幹雄(2007). 全文検索システム Hyper Estraier，fallabs.com/hyperestraier/

工藤拓(2006). MeCab ： Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

狩野芳伸(2014). 大学入試センター試験歴史科目の自動解答. 2014 年度人工知能学会全国大会(第 28 回). 愛媛県ひめぎんホール. 2014 年 5 月 13 日