

仏典全文検索システムの構築と評価

村川 猛彦¹ 丁 敏² 中川 優¹

¹和歌山大学 システム工学部 ²和歌山大学 大学院 システム工学研究科

經典撮影画像と文字情報を対応付けた、經典読解支援システムを構築するにあたり、劣化や字体の違いなどのため、機械的文字認識のみでは文字の特定が困難なものもある。本研究では、經典画像から対応するテキストファイルの特定を支援する全文検索システムを構築した。CBETAの大正新脩大藏經テキスト情報をデータベースに登録し、検索エンジン Senna の持つ近傍検索を用いて、ワイルドカード検索や複数行検索ができるようにした。16枚の經典画像に基づくテキストデータに対して、その一部を取り出して検索することで元のテキストファイルが特定できるか実験を行い、3文字程度でも複数の語があれば特定しやすくなることを確認した。

Construction and evaluation of full-text search system for Buddhist sutras

Takehiko Murakawa¹ Ding Min² Masaru Nakagawa¹

¹Faculty of Systems Engineering, Wakayama University

²Graduate School of Systems Engineering, Wakayama University

In constructing a reading support system which contrasts the shot images of some Buddhist sutra with text data, we have difficulty identifying some characters only using automatic character recognition because of sutra's degradation and of the difference of the fonts. In this paper, we report our full-text search system for Buddhist sutras. The system stores the text files derived from CBETA's Taisho Tripitaka data and provides a search form which enables a wildcard search and a multiple line search using the neighborhood search supplied by the search engine Senna. We took experiment of file identification with 16 shot images to show that a couple of trigrams often retrieve the text file.

1. まえがき

天野山金剛寺所蔵の4,500巻以上からなる一切經（金剛寺一切經）について、悉皆調査とデジタルカメラによる撮影がなされている。大正新脩大藏經との照合により、異本や新出經典を発見するとともに、漢訳仏典における奈良平安古写經の歴史的な位置付けが確立しつつある[1]。それと前後して、金剛寺一切經を含む8つの一切經に関する対照目録[2]が刊行され、国内古写經の存欠状況を一望することが可能となっている。

筆者らは、金剛寺一切經撮影画像を対象として、既存のテキスト情報と対応付け、計算機上で対照表示するための機械的手法の確立を目指している[3,4]。画像は視認性に優れているが、そのままでは検索は難しい。經典、行あるいは文字の単位で、画像とテキストを対応付けられればよいが、經典の劣化や字体の違いなどにより、機械的文字認識のみでは特定が困難なものもある。

図1は、『摩訶般若波羅蜜放光經放光品第一』の撮影画像の一部である。ここで「如光如」と「如化」が認識できたとき、それらを含むテキストファイルを求め、間の文字を判別できるようにしたい。本稿では、既発表の全文検索システム[4]よりも詳細な性能評価について述べるとともに、ワイルドカード検索などを可能とする、新たな仏典全文検索システムについて報告する。

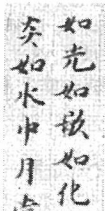


図1: 經典画像の例

2. 対象とするテキストファイルおよびその処理

中華電子仏典協会が提供する「CBETA Chinese Electronic Tripitaka Collection Feb. 2007」のCDイメージ(<http://www.cbeta.org/iso/cbeta200702.iso>)を使用し、文献[4]と同様の方法でテキスト情報の抽出を

行った。まず、CD イメージに格納されている T01.7z から T85.7z までのファイルを伸張し、大正新脩大藏經の XML 文書ファイルを取り出した。次に、iconv を用いて文字コードを Big5 から UTF-8 に変換した後に、自作の変換プログラムを実行して不要なタグや記号類を削除し、テキストファイルを作成した。その際、外字の実体参照は、ゲタ記号 (=) に置き換えた。このようにして生成された 8,989 個のテキストファイルを、順に全文検索用データベースに登録した。全文検索エンジンには Senna[5]を採用し、インデックス生成では N-gram 方式を使用している。

以前は 2005 年 2 月に配布された CD-ROM に基づいてファイルを生じていた。ここでその違いを述べる。まず、XML 文書の圧縮方法が前回は zip であったが、今回はより圧縮率の高い 7z であった。文書中で使用されている記号類が増えており、そのため変換プログラムには削除ルールを多数追加した。内容では、『放光般若經卷第一』のテキストファイル (T08n0221_001.txt) を比較したところ、2 文字だけ異なっていることを確認した。

3. ファイル特定に関する評価実験

經典画像に対して、目視や、オフライン文字認識ソフトウェアの適用などにより、連続する何文字か（「文字列」または「語」と呼ぶ）が認識できたとする。その文字列を、構築した全文検索システムで検索し、該当件数がちょうど 1 になれば、テキストファイルが特定できたことになる（内容の確認は別途行う）。この意味でファイルを特定するために、画像中の何文字を与えればよいかについて、詳細な評価実験を行った。

対象とした經典を表 1 に示す。金剛寺一切經撮影画像のうち、大般若波羅蜜多經卷第二から同第十まで、および大般若波羅蜜多經以外からランダムに 7 巻選んだ計 16 巻である。（大般若波羅蜜多經卷第一についてもテキストデータを作成したが、それを含むテキストファイルが複数あるため、対象から除外している。）各經典は卷子本であり、複数のコマに分けて撮影されているが、經卷名が含まれている先頭のコマのみを用いた。

表 1: 評価用テキストデータ

画像ファイル名	經卷名	CBETA テキスト	行数	字数	書換え
A0001-002a-03	大般若波羅蜜多經卷第二	T05n0220_002	8	115	なし
A0001-003a-03	大般若波羅蜜多經卷第三	T05n0220_003	11	166	なし
A0001-004a-03	大般若波羅蜜多經卷第四	T05n0220_004	7	98	なし
A0001-005a-04	大般若波羅蜜多經卷第五	T05n0220_005	27	438	なし
A0001-006a-03	大般若波羅蜜多經卷第六	T05n0220_006	10	149	なし
A0001-007a-03	大般若波羅蜜多經卷第七	T05n0220_007	6	81	なし
A0001-008a-03	大般若波羅蜜多經卷第八	T05n0220_008	12	183	なし
A0001-009a-03	大般若波羅蜜多經卷第九	T05n0220_009	11	166	なし
A0001-010a-03	大般若波羅蜜多經卷第十	T05n0220_010	14	216	なし
A0002-001-02	摩訶般若波羅蜜放光經放光品第一	T08n0221_001	22	365	あり
A0003-002-03	摩訶般若波羅蜜經相應品第三	T25n1509_035	24	409	なし
A0003-036-03	摩訶般若波羅蜜經善達品第七十八	T25n1509_089	28	479	あり
A0008-001-03	摩訶般若波羅蜜道行經道行品第一	T08n0224_001	24	408	あり
A0032-001-03	大寶積經序	T11n0310_001	12	204	なし
A0032-002-03	大寶積經三律儀會第一之二	T11n0310_002	13	222	あり
A0032-065-03	大寶積經菩薩見實會第十六之五	T11n0310_065	10	163	なし

評価実験の前に、前節の全文検索データベースも活用しながら、画像に対応するテキストデータ（この実験のために使用するファイルを「テキストデータ」と呼び、全文検索データベースに登録した「テキストファイル」と区別する）を作成した。古写經と、大正新脩大藏經の底本（刊本一切經）との異同も指摘されている[1,6]が、今回使用する經典画像についてはいずれも、CBETA のデータの中に該当するテキスト情報を発見できた。内容については、いくつか注意を要するものがあった。

- A0002-001-02（『摩訶般若波羅蜜放光經放光品第一』）では、表題にあって、テキストファイルにはなかった文字列があった。本文でも、いくつかの文字を追加した。該当部分を図 2 に示す。下線は、テキストファイルになく、手作業で追加した文字である。
- A0003-002-03（『摩訶般若波羅蜜經相應品第三』）では、画像の字句が出現するテキストファイルは一つに特定できたが、その出現位置は、ファイルの中の 4 箇所に分断されていた。タグ等除去前の XML 文書を参照したところ、画像の本文最初は「【經】」に続く「佛告舍利弗。菩薩

摩訶薩行般若波羅蜜」からであり、XML 文書で「【論】」と次の「【經】」で挟まれた部分（3 箇所）を飛ばすことで、画像とテキストファイルとの対応が得られた。

- A0003-036-03（『摩訶般若波羅蜜經善達品第七十八』）では、画像では「六道」、テキストファイルでは「五道」となっているのが 3 箇所あったので、「六道」に書き換えた。そのほか、左下の文字は「道」と書かれ、その右に「是」が添えられている。テキストファイルと照合したところ、この文字は正しくは「是」と推測できるが、我々は将来的には、機械的文字認識の結果をもとに検索できることを目指しており、画像の本文に合わせて「道」を採用した。
- その他にも 2 点、画像とテキストデータで一致しない文字があったが、画像で読み取れた文字を記載している。また、経典に穴が開いており画像では判別できなかった箇所については、テキストデータの記述をそのまま使用した。テキストファイルから取り出して、字句を書換えたものを、表 1 最右列で「あり」、書換えていないものを「なし」としている。

三藏無羅叉奉詔譯
摩訶般若波羅蜜放光經放光品第一
聞如是一時佛在羅閱祇耆闍崛山中與大
比丘眾五千人俱皆是阿羅漢諸漏已盡意
解無垢眾智自在已了眾事譬如大龍所
作已辦離於重擔速得所願三處已盡正解
已解復有五百比丘尼諸優婆塞優婆夷諸
諸菩薩摩訶薩已得陀=尼空行三昧無相無

図 2: 経典テキストの例

評価実験では、この各テキストデータに対して、連続する N 文字を取り出し ($1 \leq N \leq 17$, 複数行にまたがらないものとする)、これを「認識できた文字列」とみなして一つずつ全文検索システムに与え、該当件数が 1 になるかを調べた。ここで語長 N の最大値を 17 としたのは、今回の経典画像は、表題等を除き基本的に 1 行 17 字で書かれていたためである。文字列の字数とファイル特定率の関係を図 3 に示す。細線が各経典、太線は平均である。この図から、以下のことが分かる。

- 語長が 1 のとき、すなわち単一の文字では、テキストファイルを特定できなかった。
- 語長が大きくなるにつれ、ファイル特定率も向上する。例外もあるが、A0001-007-03 については、「初分相應品第三之四」で構成される行の末尾 6 文字でのみ特定できており、語長が 10 以上になれば、この行が対象外となるために特定率が 0 となっている。それ以外の低下は語長が大きくなったときに見られ、前述の通り書換えたのが影響している。
- 語長を大きくした際のファイル特定率の伸びは、テキストデータによって大きく変わる。これはテキストデータの行数や字数、あるいは大般若経であるか否かに依存しない。
- 語長を大きくした際のファイル特定率の伸びは、おおむね語長 3 から 6 まですでに決まる。ただし A0001-010a-03 のような、語長 8 あたりから伸び始めるような例外もある。

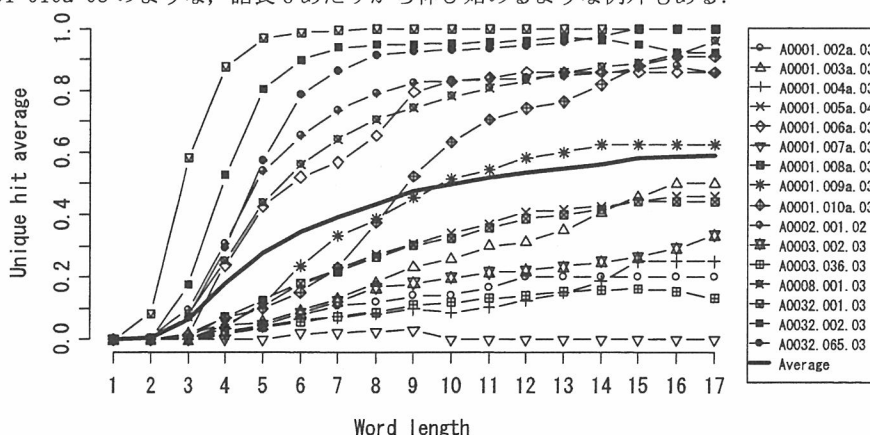


図 3: 語長に対するファイル特定率

なお、該当件数 1 のテキストファイルはいずれも、事前作業で発見したものと同一であった。

該当件数 1 にこだわらず、件数が少なければ、目視または何らかの方法でテキストファイルを特定できないかという観点で、語長と、各語の該当文書数の関係を、經典ごとに求めてみた。図 3 で伸びの早い經典は、5 文字程度でも特定できるが、平均で見ると、3 文字での平均該当文書数は 500 以上あり、6 文字で 100 を切る程度だった。平均して 10 件未満になるには、12 文字程度が必要となることも分かった。

ところで、画像を文字認識し、得られた文字列を全文検索システムに問い合わせでテキストファイルを獲得して、未認識文字を求めるというアプローチをとる場合、連続して長い字数を認識する必要はないし、技術的な困難が伴う。そうではなく、經典画像の中で正しく認識できたのが 2~4 文字で複数箇所ある場合に、それらをすべて含むテキストファイルを特定できるのではないかという方針で、新たな実験を行った。

具体的には、各テキストデータからランダムかつ重ならないように語長 n の文字列を m 個取り出し ($2 \leq n \leq 4$, $2 \leq m \leq 3$)、AND 検索を試みた。テキストデータあたり 1000 回実施し、ファイル特定率の平均を求めた。さらに、前述の実験 (単一検索語に関する語長とファイル特定率の関係) と比較するため、以下に定義する換算語長を求めた。

換算語長: あるテキストデータに対して、語長 n ($n=1, 2, \dots, 17$) の単一検索語に対する平均ファイル特定率を $a[n]$ と書く。複数検索語に対する平均ファイル特定率が α のとき、 $a[k] \leq \alpha \leq a[k+1]$ を満たす最小の整数 k を求め、

$$k + (\alpha - a[k]) / (a[k+1] - a[k])$$

を、その複数検索語における換算語長という。

この式は、平均ファイル特定率 α が、 $a[1], \dots, a[17]$ の間にあるとき、線形補間を用いて、単一検索語に相当する語長を求めるものである。なお、 $\alpha=0$ または $\alpha < a[1]$ のときは除外する。任意の k に対して $\alpha > a[k]$ となる場合には、単一検索語よりもファイル特定の性能が高いことを意味する。

上のようにして求めたファイル特定率および換算語長を、表 2 および表 3 に示す。換算語長の平均は、単一検索語よりも特定率が低い場合 (表 3 の「-」の位置) は除外し、高い場合 (同「>17」) は 17 として、求めている。

いずれの場合も、同じ語長の単一語検索よりも良い特定率となっている。また $(n, m) = (3, 3), (4, 2), (4, 3)$ のとき、換算語長は $n \times m$ よりも大きくなった。これは、長さ $n \times m$ の単一文字列よりも、ファイル特定率が高いことを表し、3 文字程度でも複数箇所での認識ができれば、特定が可能なことを示唆する。

表 2: 複数語検索におけるファイル特定率

經典画像 ファイル名	(語長, 語数)					
	(2, 2)	(2, 3)	(3, 2)	(3, 3)	(4, 2)	(4, 3)
A0001-002a-03	0.007	0.023	0.039	0.146	0.125	0.247
A0001-003a-03	0.009	0.070	0.179	0.339	0.233	0.494
A0001-004a-03	0.007	0.033	0.074	0.207	0.124	0.295
A0001-005a-04	0.000	0.002	0.049	0.148	0.185	0.324
A0001-006a-03	0.001	0.025	0.193	0.383	0.433	0.624
A0001-007a-03	0.000	0.000	0.029	0.094	0.047	0.151
A0001-008a-03	0.001	0.017	0.178	0.344	0.293	0.528
A0001-009a-03	0.000	0.032	0.126	0.338	0.268	0.494
A0001-010a-03	0.005	0.036	0.124	0.302	0.269	0.476
A0002-001-02	0.067	0.204	0.547	0.842	0.849	0.918
A0003-002-03	0.000	0.005	0.060	0.138	0.159	0.293
A0003-036-03	0.000	0.001	0.021	0.061	0.080	0.107
A0008-001-03	0.009	0.061	0.401	0.669	0.695	0.899
A0032-001-03	0.439	0.780	0.979	1.000	0.999	1.000
A0032-002-03	0.116	0.298	0.739	0.937	0.947	0.963
A0032-065-03	0.014	0.084	0.461	0.809	0.862	0.986
平均	0.042	0.104	0.262	0.422	0.411	0.550

表 3: 複数語検索における換算語長

経典画像 ファイル名	(語長, 語数)					
	(2, 2)	(2, 3)	(3, 2)	(3, 3)	(4, 2)	(4, 3)
A0001-002a-03	3.315	4.029	4.632	10.132	8.223	>17
A0001-003a-03	2.429	5.368	7.944	12.707	8.985	15.856
A0001-004a-03	3.532	4.654	7.120	14.312	11.954	>17
A0001-005a-04	-	2.383	4.020	5.556	6.133	9.502
A0001-006a-03	2.064	3.042	3.800	4.772	5.075	7.652
A0001-007a-03	-	-	8.544	>17	>17	>17
A0001-008a-03	2.158	3.155	5.956	10.561	8.748	>17
A0001-009a-03	-	3.845	5.085	7.088	6.326	9.651
A0001-010a-03	2.312	3.374	5.457	7.496	7.272	8.674
A0002-001-02	2.672	3.506	5.063	12.325	13.307	>17
A0003-002-03	-	3.210	5.393	7.369	7.831	16.032
A0003-036-03	-	3.066	4.251	6.122	7.402	8.850
A0008-001-03	2.120	2.811	4.788	7.409	7.812	15.424
A0032-001-03	2.714	3.679	5.616	8.000	7.869	8.000
A0032-002-03	2.656	3.350	4.763	6.998	8.330	12.190
A0032-065-03	2.199	3.060	4.595	6.302	6.990	14.496
平均	2.561	3.502	5.439	9.009	8.704	13.208

4. 仏典全文検索インタフェース

より柔軟な検索を対話的に実施できるよう、検索インタフェースを構築した。この検索システムでは、単純なフレーズ検索のほかに、字体の違いを考慮した検索と、ワイルドカード検索、複数行検索を実現している。図 4 はその画面例であり、図 1 の未判別文字を「影」と推定できる。

入力

如光如? 如化

検索 ☐ 字体変換しない

検索結果

検索語: 如光如 如化

該当件数: 1

順位: 1

ファイル名: T08n0221_001.txt

経典名: 放光般若經卷第一

適箇所: 堪任教化所説如幻如夢如響 如光如影 如化如水

テキストデータ

ファイル名: T08n0221_001.txt [\[別画面\]](#) [\[XML\]](#)

図 4: 仏典全文検索システムを用いた検索例

ユーザ側画面は、検索語入力フォーム、検索結果、経典テキストデータからなる。該当するものがあれば、検索結果の欄には内部スコアが 1 位のファイル名、経典名（テキストファイルの最初の行）、KWIC 形式で適箇所（最大 3 箇所まで）を表示し、テキストデータの欄には、17 文字で折り返したテキストファイルを表示する。このテキストファイルをブラウザの新規ウィンドウで表示したり、元の XML 文書を表示したりすることもできるよう、リンクをつけている。該当するものが複数あれば、検索結果の順位の行に「次へ」や「前へ」というリンクがつく。これにより、他の順位のファイルへ移ることができる。該当する文書がない場合には、文字単位の出現文書数を返す。入力ミスをした文字が分かります。

Ajax の技術を活用して、ページの遷移をしないようにしている。ブラウザで検索と動作確認ができるインタフェースとしたが、クライアントは HTTP を通信するものなら何でもよく、所定の URI でアクセスすれば、サーバは該当件数とテキストファイルの情報を含む XML 文書を生成して返す。

サーバ構築では、CGI プログラムを Ruby で自作した。その処理は、検索語から全文検索エンジン向けの検索式への変換、全文検索エンジンへの問い合わせ、問い合わせ結果をもとにユーザ側へ返す情報の生成に大別される。本稿では、検索式の変換に絞って説明する。

与えられた検索語に対して、句読点等の記号を除去してから、サーバ内に持つ字体変換テーブルに基づき字体を変換する。例えば、検索語が「供養如来及聞授」であれば、「供養如来及聞授」に変更する。字体変換テーブルの詳細は次節で述べる。

ワイルドカード検索では、例えば「如光如？如化」を検索語としたとき、「如光如」と「如化」の2語に分割してから、Senna が提供する近傍検索を使用するため、「*N1 “如光如 如化”」という検索式を生成する。

さらに、文献[4]の横方向検索に対応する試みとして、複数行検索ができるようにした。本研究では、インデックスを単一とし、検索式を工夫することで解決を図った。図1の先頭2文字ずつを取り出し、「如光／炎如」を検索語とすると、サーバ内部ではSenna への検索式「*D+ *N16 “如光 炎如”-(*N13 “如光 炎如”）」に変換する。これにより、「如光」と「炎如」を含み、その距離（間の字数）が13文字を超え16文字以下であるようなテキストファイルを求める。

5. 字体の変換

筆者らの先行研究[4]で、課題の一つとして指摘したのが、新字体検索である。本システムでは、与えられた検索語に対して、サーバ内に持つ字体変換テーブルに基づき字体を変換することを試みた。検索する際には新旧いずれの字体を使用してもよい。結果を参照し、必要に応じて原文の一部を取り出す際には、繁体字のままがよいと考えた。

字体変換テーブルの構成法について述べる。ユーザが自由に字体変換ルールを登録できるようにすれば、そのルールの充実に応じて、スムーズに検索できるようになるであろう。しかしその方式は、初期状態では字体変換ができないことを意味する。また不適切な変換ルールが登録された場合の対処も管理上の手間となる。そこで本研究では、既存の字体変換ルールをもとに、前計算して取捨選択し、検索システムに組み込むことにした。

既存の字体変換ルールとして、(1)新旧字体表[7]、(2)山田ら[8]の示すコードセパレート文字を使用した。それらに記載されている各文字に対して、2節の全文検索データベースを用いて、CBETA 文書での出現状況を調査した。結果を表4に示す。コードセパレート文字において Big5 収録の文字についても、便宜上、旧字体としている。

表 4: 新旧字体の使用状況

旧字体の 出現文書数	新字体の出現文書数			
	0 件	1 件	2 件以上	合計
0 件	10	0	19	29
2 件以上	298	3	83	384
合計	308	3	102	413

新字体では出現せず、旧字体では出現するような文字の組については、新字体から旧字体に変換するルールを追加する。新字体で2件以上出現するものについては、旧字体で多数出現するものであっても、変換しないこととする。新旧字体の例でよく知られる「弁」については、「弁」が37件、「辨」は1829件、「辯」は39件、「辯」は3639件であった。なお、基本漢字外の中で、「逸」とその旧字体（ユニコード FA67）など75組については、出現文書数が（出現文書も）同一であったが、それらは変換してもしなくても同じ文書が得られるという意味であり、ルールには追加しなかった。

新字体で1件のみ出現する文字があった。具体的には「与」（旧字体は「與」）、「体」（同「體」）、「証」（同「證」）である。これらについては、旧字体のほうが数千の文書で使用されていることと、いずれの組についても、該当する新字体が出現する文書に、その旧字体も含まれている（例えば T51n2073_001 には、「…體甚康体…」と、近い距離に両方の文字が出現している）ことを考慮し、これについても新字体を旧字体に変換するルールとした。

なお、旧字体で出現文書数がちょうど1となった文字は見つからなかった。2件については、「槓」が2件、その旧字体の「槿」が0件であった。また、コードセパレート文字については、いずれも新字体は出現せず、旧字体（繁体字）が1件以上出現していた。

このようにして得られた字体変換ルールに対して、Ruby において変換対象文字を検出するための正規表現も生成しておく。検索語が与えられたとき、gsub メソッドを用いてパターンマッチを試み、該当する文字をそれぞれ置換している。検索語の字数が小さいため、処理は一瞬で行われる。

図4の入力フォームにあるように、字体変換をしないよう指定することも可能である。

6. 仏典全文検索システムの評価

経典画像を見ながら、いくつかの検索語を与えて該当件数を求めた。結果を表 5 に示す。「葉当??時皆」のように、間に 2 文字の未判別文字があっても特定できる場合があることを確認した。

最後の例は、文献[4]でも検索できなかった例である。Senna への検索式の一部「-(N14 “生 真”)」により、期待するテキストファイルも排除されたものと思われる。また「如?光影如化」については、「如」と「光影如化」がともに出現するが、その距離が 1 を大きく離れるテキストファイルも取得している。他にも、ワイルドカード検索や複数行検索で、語長が 1 のものを含むと、期待する検索結果にならなかった。

表 5: 仏典全文検索システムによる検索語例

検索語	該当件数	備考
如光如?如化	1	
如光如 如化	8	
如?光影如化	8	
如光/炎如	1	
如光 炎如	5	
葉当??時皆	1	当⇒當
南/生/真/光	0	

ワイルドカード検索について、網羅的な評価を行った。3 節で使用した 16 のテキストデータを使用し、行をまたがらない 5-gram の中央の文字を取り除いた 2 文字 2 語で AND 検索するものを「ワイルドカード検索なし」、同様に 5-gram の中央の文字を「?」に置き換えた文字列で検索するものを「ワイルドカード検索あり」として、ファイル特定率の平均と換算語長を求めた。換算語長は（ファイル特定率が 0 で、除外される場合を除き）2 が下限、5 が上限となる。結果を表 6 に示す。これにより、ワイルドカード検索が特定のしやすさに貢献することを確認できた。

表 6: ワイルドカード検索のファイル特定率および換算語長

	ワイルドカード検索	
	なし	あり
ファイル特定率（平均）	0.039	0.210
換算語長（平均）	2.705	4.312
特定できなかったテキストデータの数	9	1

7. 関連研究および関連システム

大正新脩大藏經の全文検索サイトとして、CBETA サイト[9]と広済寺の仏教典籍検索[10]が知られているが、その特徴は[4]で述べた通りである。すなわち、CBETA サイトの全文検索は Google に依拠しており、経典テキストの文字コードは Big5（繁体字）である。広済寺サイトでは全文検索エンジンに Namazu を用い、KAKASI および独自の仏教用語辞書により分かち書きをしている。データは新字体で Shift_JIS である。ともに、検索において利用者は一方の字体の利用を強いられるが、本研究ではデータは繁体字のままとし、検索語の入力に対して内部で字体を変換している。字体変換の可否は、チェックボックスで設定できる。これにより、より柔軟な検索システムを構築できた。

文献[4]では、全文検索エンジンに Hyper Estraier を用いて検索システムを構築してきた。そこでの課題として、いわゆるワイルドカード検索ができないことが挙げられる。Hyper Estraier 自体は正規表現検索の機能を持つが、英単語に限られる。そこで筆者らは、最近脚光を浴びている Senna の持つ近傍検索機能に興味を持ち、検索エンジンの変更を試みた。

ホームページ[5]によると、Senna は、DBMS（データベース管理システム）などに組み込んで全文検索機能を提供するものである。DBMS に PostgreSQL を用いたものは Ludia[11]、MySQL を用いたものは Tritonn[12] として、いずれもフリーで利用できる。本研究では Tritonn を用いている。バージョンは、senna-1.0.9 および mysql-5.0.45-tritonn-1.0.6 である。

検索エンジンの移行に当たり、結果が異なっていれば、そして Senna のほうが悪いものであれば、移行は躊躇せざるを得ない。そこで、Hyper Estraier と Senna とで検索結果を比べことにした。登録対象のテキストファイルは、2 節で作成した 8,989 個のテキストファイルである。Hyper Estraier のバージョンは 1.4.10 で、文書登録時に、N-gram 法で処理するオプションを付与した。

その結果、検索語長が 2 のときは、「ニ」を含むもののみ異なり、それ以外はすべて一致した。検索語長が 2 以外のときは、違いが多く見られた。

Ruby を用いて全検索により出現文書数を求めるプログラム（検索エンジン不使用版）を書き、違いが見られたいくつかの検索語に対して実行して、Hyper Estraier および Senna の結果と比較したところ、検索エンジン不使用版と Senna との出力が一致した。ここから、Hyper Estraier よりも Senna のほうが正確であるといえる。

検索語長が 2 のときに一致し、それ以外では違いが見られた原因として、Hyper Estraier, Senna とも、テキストファイルや検索語を bigram で切り出ていることが指摘できる。該当文書を瞬時に求める際、2 文字については全文検索インデックスから直ちに得られるが、1 文字や、3 文字以上においては、全文検索インデックスの bigram 情報を組み合わせる必要がある。ここに全文検索エンジンの個性が出たとともに、6 節の、1 文字の語を含む近傍検索がうまくいかない遠因とも考えられる。

ワイルドカード検索とは別に、接続確率を保持した bigram 統計情報を使って、和文全文検索の精度（再現率）を向上する試みが太田ら[13]によってなされている。

8. あとがき

本稿では、CBETA の経典テキストデータに対して構築した全文検索システムと、それを用いたテキストファイル特定評価実験について述べた。その成果は以下の 3 点に集約できる。

- 16 枚の経典撮影画像からテキストデータを抽出し、連続する N 文字を検索語として、ファイル特定率を求めた。経典（画像）から認識できる文字が対応するテキストファイルと 1 文字でも異なっていると、ファイルを発見できないことを、N の増加に対して平均特定率が下がるいくつかの現象から確認できた。
- 経典画像から連続できた文字列の長さが 2~4 程度であっても、複数箇所あれば、AND 検索によりテキストファイルを特定しやすくなることを確認した。
- 全文検索エンジンの近傍検索機能を用いて、ワイルドカード検索や複数行検索を実現した。これらにより、単なる AND 検索では特定できなかったものも特定できるようになった。ただし 1 文字の語を含む検索は、期待通りに機能していないことも分かった。

本研究では、ファイルを特定する（該当件数が 1 になる）方法を模索してきたが、本検索システムの用途はこれにとどまらない。例えば小字数の AND 検索やワイルドカード検索は、異本テキストの発見にも有用である。一部の文字を隠して、該当する字句がどれだけあるかを検出するという、より高機能な全文検索として使うことも可能である。

謝辞 本研究を進めるに当たり、様々なご教示を賜った、国際仏教学大学院大学の落合俊典教授および青木進氏に心より感謝いたします。

参考文献

- [1] 金剛寺一切経の総合的研究と金剛寺聖教の基礎的研究, 平成 16~18 年度科学研究費補助金基盤研究(A)研究成果報告書, 研究代表者 落合俊典, 2007.
- [2] 国際仏教学大学院大学 学術フロンティア実行委員会: 日本現存八種一切経対照目録, 331p., 2006.
- [3] 張蓉, 仁野洋平, 田中猛彦, 中川優, 青木進, 宇都宮啓吾, 落合俊典: 仏典データベースのための画像処理について, 情報処理学会研究報告 2006-CH-69, pp.25-32, 2006.
- [4] 田中猛彦, 仁野洋平, 中川優: 仏典データベースのためのテキスト処理について, 情報処理学会研究報告 2006-CH-69, pp.33-40, 2006.
- [5] Senna 組み込み型全文検索エンジン, <http://qwik.jp/senna/FrontPageJ.html>
- [6] 落合俊典: 二種の『馬鳴菩薩傳』—その成立と流傳—, 七寺古逸經典研究叢書第五卷 中国日本撰述經典 (其之五)・撰述書, 大東出版社, pp.619-646, 2000.
- [7] 新旧字体表 http://www.asahi-net.or.jp/~ax2s-kmtm/ref/old_chara.html
- [8] 山田崇仁, 小島浩之: データベースナビゲーター, 漢字文献情報処理研究, Vol.6, pp.65-74, 2005.
- [9] CBETA 中華電子佛典協會, <http://www.cbeta.org/>
- [10] 仏教典籍検索, <http://www.kosaiji.org/~kyoten/>
- [11] Ludia, <http://www.nttdata.co.jp/services/ludia/index.html>
- [12] Tritonn プロジェクト, <http://qwik.jp/tritonn/>
- [13] 太田学, 高須淳宏, 安達淳: 認識誤りを含む和文テキストにおける全文検索手法, 情報処理学会論文誌, Vol.39, No.3, pp.625-635, 1998.