大谷大学所蔵西蔵文献目録のオンライン
データベース化について

齊藤友啓

目　次

――大谷大学所蔵西蔵文献目録のオンラインデータベース化について――

1　序論

(1) はじめに

私は、福田ゼミに在籍しプログロミング言語やチベットについて学習してきた。そして、チベット文化の発展に役立ちたいと考えるようになり、その考えとゼミの「人の役に立つ Web アプリケーションを作ろう」というテーマが合わさった結果、卒業論文のテーマは、「大谷大学所蔵西蔵文献目録のオンラインデータベース化について」に至った。

(2) 現状と目標

大谷大学には、貴重なチベット文献が数多く所蔵されている。その多くの文献の目録は、出版され世界中で閲覧できる。しかし、膨大な量の目録から一つ一つ本を手にしながら必要な文献情報を探し出すのは容易ではない。さらに出版されているとはいえ、一部の大学にのみ置かれているのが現状であり、容易にチベット文献の情報を得ることができない。その現状を打破するために西蔵文献目録の検索システムの構築を目指す。昨今では、unicode の普及によりチベット文字を表示・入力できる環境が増加した。そのため、検索システムには unicode を使用しチベット文字を用いて検索、表示を行えるようにする。そして、最終的にはインタネット・サーバー上で稼働させることが目標である。

(i) 誰の役に立つのか

チベット仏教研究者又は、チベット文献を必要としている方々である。また、今回の制作を通して得た知識を蓄えることで、今後のチベット文字を用いたデータベース作成の手がかりになると考える。

（ii）大蔵経と蔵外文献

チベット文字で書かれた文献は、インドの仏典を翻訳したものと、チベット人が書いた著作を集めたものに分けられる。インドの翻訳仏典を集めて集大成したものは大蔵経と呼ばれ、それ以外のチベット人が書いたものは蔵外文献と呼ばれる。今回、検索システムを制作する基になった目録の内容は、１９７３年に大谷大学に所蔵されていた、大蔵経以外のインドの翻訳仏典と蔵外文献である。そのうち、翻訳仏典は大蔵経に含まれるので、この目録の価値は蔵外文献にある。そのため、テーマは、西蔵文献目録としているが実際は、蔵外文献目録である。（以下、蔵外文献と呼ぶ）

2　制作するにあたり

(1) 参考にしたサイト

本学では、北京版チベット大蔵経目録検索サイト(1) が制作されており目録検索を行うことができる。しかし、制作された当時は、ＯＳの問題でチベット文字を表示させる環境がなく、やむを得ずローマ字表記で表示させなければならなかった。さらに、検索結果の表示をページ送りもせずに一括で表示されるなどの不都合な点があり、２００７年度卒業生の美濃部春菜さんの手により新しい北京版チベット大蔵経目録検索サイト(2)（以下、新北京版と呼ぶ）が制作された。新北京版では、チベット文字での検索、表示が可能になり、データベースを使用し、項目別に検索できるなど完成度が高い。新しく検索システムを作る上で大いに参考になると判断したため、蔵外文献目録検索システム（以下、蔵外検索と呼ぶ）では、新北京版を踏襲することとした。

(2) 新北京版の改善点

新北京版（資料編、図５参照）を踏襲するうえで、機能の改善を図る点を模索した。以下、私が実際に新北京版を操作して感じた点である。

(i) 表示の大きさ

まず、気づいたのが新北京版の文字サイズが全体的に小さい点である。見づらいという印象を受けたとともに、検索語句の入力フォームが非常に小さく作られており、この点は、改善しなけらばならないと考えた。なぜなら、入力するチベット文字は、非常に複雑な構造を持っているからである。チベット文字は、１つの単語をいくつかの文字が合わさって表す。そして、結合する際には、横並びに結合するだけでなく、縦にも結合して文字の形を変化させる。その結果、結合するたびに複雑な形になり文字が小さくなるので非常に見づらくなってしまう。ブラウザの機能で文字サイズを変更すれば済むことではあるが毎回、設定するのは面倒である。そのため、表示する文字のサイズ、入力フォームのサイズを CSS ファイルで font-size を定義し、（資料編、zogai.css、４行目、２９行目参照）新北京版より数段大きく表示させるようにする。

(ii) 表示件数の誤り

新北京版では、検索結果が２０件以上だった場合、２０件までのデータを表示し、残りの結果を次のページに送る、ページ送りシステムを搭載している。これは、一つのページに全ての検索結果を表示しないことで検索処理や表示の速度を向上させている。しかし、常に結果表示が、２０件に固定されているので、一度に大量のデータを見たい方や、逐一ページ送りをしたくない方には使いづらいと考える。そこで、ユーザーの自由意思を尊重するためにある程度は、表示件数を選べるように変更する。また、最後のページの表示件数が残り１２件であっても、リンク

の表示が、NEXT ２０と表示されてしまうので、残りの件数を計算してNEXT １２と正しく表示されるよう改善する。(資料編、図６、資料２参照)

（iii）検索結果の表示色

北京版チベット大蔵経目録検索サイトから受け継いでいる「検索結果で表示されているデータの中から、検索した語句があればその語句だけ赤色で表示する」という機能は、そのまま蔵外検索でも採用することにした。この機能は、どの部分が検索した語句なのか容易にわかるため、非常に便利である。（資料編、図６参照）

（iv）福田先生の指摘

蔵外検索を制作するうえで、検索者が検索サイトを初めて見た時、どのようなデータを得られるのか、また調べられるのか、解り易くして欲しいとの意見を頂いた。新北京版では、検索項目をプルダウン形式で選び、その項目内から入力された語句を検索できるようになっている。（資料編、図５参照）蔵外検索では、項目を選択するプルダウン形式を廃止し、一目で何の項目から検索できるのか解るように変更する。

また、検索する際にどの項目内から検索を行うかプルダウン形式で選択する必要があるが、この方法では、例えばタイトルを選択するとタイトルの項目内のみ検索を行うので、タイトルと著書の項目といった複数の項目で同時に検索ができない点がある。より検索者の思惑通りのデータが抜き出せるよう、複数項目での同時検索を行えるよう改善する。

以上の点に注意し、検索者が利用しやすい蔵外文献目録検索システムの構築を目指す。

(3) 使用した技術

今回、使用する蔵外のデータは、テキストデータで保存されている。しかし、テキストデータのままでは、データの管理、検索が不便である。また、データの追記・修正を行う際に非常に手間がかかる。効率の良い検索システム構築のため、テキストデータは、データベース化を行う。データベースを構築、管理するソフトとして MySQL を利用した。MySQL を外部から操作するために PHP、ブラウザ表示に HTML、表示するデザインを指定するために CSS を使用した。また、データベースに関する基本知識と操作を学ぶため『MySQL 入門以前』を用い知識を深めた。

3　本論

(1) 文献のデータベース化

(i) データベース化の利点

蔵外文献のデータは、テキストデータで構成されているため、データベース化する必要があると考える。データベース化には様々な利点がある。

まずデータベース化することにより、データの把握が容易に出来る。蔵外のテキストデータは、上から順にタイトル、著者名（他の項目が続く）というように並んでいると最初に宣言し、その下から、一件分ずつデータが羅列されているだけで、（資料編、図８参照）一見しただけでは、何がどのデータなのか判別しづらい。そのうえ、羅列されたデータが、３５４１件も保存されているので、データの把握に非常に時間がかかる。そのため、データベース化することで、項目別にデータを格納し、容易にデータの取り出し、確認を行えることが適当であると考える。

次に上記で述べた、データの確認が容易にできることによってデータの不備（誤字や重複して登録されているデータ）が発見できる。３５４１件ものデータが羅列されていては、どこに何が書かれているのか容易に把握できない。誤字、脱字の確認すら非常に大変である。その点、データベースでは、項目ごとに整列させることが可能であるので、例えば、タイトルだけの項目を表示させることができ、確認が容易になる。また、不備を発見した際には、修正を行える。そして、データの追加も容易に行えるようになる。追加項目を全てのデータに反映させられるので入力こそ大変であるかもしれないが、テキストデータよりは、時間がかからないと考える。

最後に検索処理の向上である。基のテキストデータは項目別に分かれていないため、検索を行う際、全てのデータの中から検索語句を検索することになる。その場合、検索処理に時間がかかり、検索者の引き出したいデータだけが思い通りに引き出せるとは到底思えない。そのため、データを項目別に分けることにより、検索者がその項目内で検索したい語句を入力することにより、より検索者の要求する結果を導き出せ、処理速度も向上すると考える。

以上の利点により、テキストデータをデータベース化する。

（ii）データベース化の過程

私は、検索システム構築を行ったが、データベース化した蔵外文献のデータは提供して頂いた。大谷大学真宗総合研究所西蔵文献研究班から頂いたデータのデータベース化の過程については、福田先生から次のように伺っている。

蔵外文献目録は、Mac のハイパーカードというソフトによってデータ化されていた。しかし、Apple がハイパーカードのサポートを終了す

ると宣言し、さらに新しい OS、MacOSX との互換性が無かったため、制作した文献データが活用できなくなってしまう可能性が浮上した。ソフトに互換性がない場合、そのソフトに対応した環境でしかデータを使用できないため、時間が経つにつれ非常に扱いにくくなってしまう。その問題を避けるためにハイパーカードから文献データを互換性のあるテキストデータに書き出し、保存されることとなった。

今回データベース化するにあたり、まず、項目順に羅列されているだけのテキストデータを一件ずつ項目ごとにタブで区切り、配列データに整列させた。（資料編、図９参照）次にタブ区切りに並べたデータを Python 言語で必要なデータだけ（データベース化して使用するデータ）取り出して並ぶようにプログラミングしたファイルで実行する。（資料編、資料３参照）

最終的に出来上がったデータが zogai_data.sql である。（資料編、図１０参照）データベースに挿入する項目とその順番に中に入るデータがプログラミングで指示した通りに並んでいる。この sql ファイルは、データベースにデータを挿入するための insert 文を記述したファイルであり、データベースを定義する sql ファイルは、別に用意している。（資料編、資料４参照）定義ファイルに直接データをいれて一つのファイルにまとめることも可能であるが、定義ファイルは、データベース作成後、追加、修正する際に確認する必要がある。そのため、データベースの定義を確認する度にデータが入った巨大なファイルを確認するのは、時間がかかるため今回は二つに分けている。

(2) サイトの構成

蔵外検索のデザインは、新北京版の構成を踏襲している。(資料編、資料１参照)トップページは目録検索に関する説明と注釈、検索する語句を入力するフォームがある。(資料編、図１参照)また、その下に kye_letter データを用いた一覧表示検索が行える表を作成している。(資料編、図３参照)実際に検索処理を行い、検索結果を表示するページは別のファイルにまとめている。また、結果表示ページからも更に絞り込み検索や、新たな検索が行えるように、上部に検索フォームを設けている。(資料編、図２、図４参照)一覧表示検索の結果と語句入力による検索結果のページは、検索の複雑化を避けるために別々のファイルに分けている。そして、レイアウトを指示する CSS ファイルを全てのページに適用させている。その他に、データデータベース作成を定義した sql ファイルとデータベースに挿入する文献データの sql ファイルを MySQL に読み込ませている。

(3) 検索方法

(i) 一致検索

検索方法は、入力された語句を含むデータを抜き出す一致検索のみ採用している。つまり、～から始まる、～で終わるといった曖昧な条件検索は採用していない。検索者は、ある程度目的を持って検索をするという考えを前程にしているため必要なデータが抜き出しやすい一致検索のみという結果になった。

(ii) 項目別検索

データベースのデータを検索する際は、検索できる項目を一覧表示し、(資料編、図１参照)その横にある入力フォームから各項目に対応

した語句を入力する形の項目別検索を採用した。一般的にフリーワード検索と呼ばれる、項目を無視して入力されたキーワードから全データ検索を行うことはできない仕様にしている。項目を選択するのは、検索語句を全データから探す手間を省くためである。

（iii）and 検索

検索を行う際に、１つの単語だけで検索することは少ないと考える。多くの人は、自分の望んだ通りの検索結果が表示されるように複数の単語を入力するだろう。そのため、検索語句の間に半角スペース、又は全角スペースが入力されれば、複数の単語で and 検索処理が行えるようにしている。その点、or 検索を採用しなかったのは多くのデータが抜き出されるより、必要なデータを的確に抜き出させることに重点を置いたためである。

例として、タイトルに何か検索語句が入力された場合、PHP での処理は以下の通りである。まず、trim 関数で入力された検索語句の先頭、および末尾に半角スペースがある場合、これを取り除く。次に全角スペースが含まれる場合は、preg_replace 関数を使い全角を半角スペースに置換する。次に explode 関数で半角スペースごとに検索語句を配列データに格納する。preg_replace 関数で全角スペースを半角スペースに置換したのは、explode 関数で検索語句をうまく配列にするためである。そして、スペースが含まれていない場合、配列には、一つのデータだけが入る。もし、スペースが入り、複数の語句が配列に格納されていれば、count 関数で配列の数を数え、その数だけ配列に格納された検索語句を抜き出し、title_words という変数に追加される。（資料編、zogai_result.php、４３～８２行目参照）

（iv）複数項目同時検索

序論の改善点より、一つの項目からしか検索できないのは、不便であり、複数の項目で同時に検索を行うことが可能であれば、より検索精度が向上すると考える。蔵外検索では、検索語句を入力する項目がタイトル、著者名、テキストナンバーの３種類のみなので、従来のプルダウン形式を廃止し、項目に合わせて入力フォームを３つ用意している。そして、入力されたフォームの項目のみ処理を行うように if 文で指示している。そして、検索語は各項目の words 変数に挿入され if と or により入力された項目の words 変数だけデータベースにアクセスする sql 文に追加される。（資料編、zogai_result.php、８４～１０１行目参照）

（v）key_letter を用いた一覧検索

key_letter というデータは、日本語でいうひらがなの５０音と同じと考えて頂いて良い。チベット文字の基本になる文字を含むデータの一覧表示である。例えば日本語でいうと、頭文字が「あ」から始まる一覧を表示するといったものになる。（ただし、チベット文字の基字は頭文字にくるとは限らない）チベット文字の基字の表をテーブルタグで作成し、基字一つずつに対応したチベット文字のデータを key という変数に代入させ、リンクを作成している。そして、一覧表示させたい基字をクリックすると結果表示のページにジャンプするとともにＧＥＴメソッド(3)によりデータが zogailist ファイルに送信され検索、結果表示を行う。（資料編、zogailist.php、２９～５８行目参照）

(4) 結果表示

(i) 表示件数の選択

新北京版では、検索結果の表示件数が２０件で固定されている。しかし、常に２０件で固定されるのは、やや不便であるので、表示件数を変更できるようにしている。form データの送信時に新たに指定された表示件数を送信する pagesize という変数を用意した。そして、select タグでセレクトボックスを作成し、１０件、２０件、５０件、１００件から指定した表示件数を変数 pagesize に入れ form で検索語句と一緒に送信するようにしている。また、デフォルト設定では結果画面を素早く表示させるために１０件に設定している。（資料編、zogai.php、４３～４９行目参照）

(ii) ページ処理

受け取ったデータを処理し、まず、検索結果の件数だけを先に求める。（資料編、zogai_result.php、７７～９４行目参照）結果の件数を把握し、それを pagesize（一ページに表示させる件数分）で割ることで、全体のページ数を求める。その際、ceil 関数を用いることによって計算結果の端数は全て切り上げられる。例えば表示件数が１０件で検索結果が２１件だった場合、計算結果は２．１になるが端数は切り上げになるので、全体のページは３ということになる。（資料編、zogai_result.php、９６行目参照）そして、最初に検索した時のみ、そのページが１ページになるように page という変数を用意し、page 変数に１を代入する。全体のページ数が２ページ以上だった場合は、次のページに移動するリンクを作成する。次のページに移動するリンクをクリックした際はＧＥＴメソッドでデータを送信する。送信するデータは検索結果の件数、検索項目に入力された語句、表示件数、表示するページの番号である。（資料

編、zogai_result.php、１２０～１３０行目参照）また、次のページに移動した際、同様に戻るページのリンクも作成している。ページ数が定義されている場合、if 文により検索結果の件数を調べる sql 文は、無視され、無駄な処理を行わないようにしている。

次に、実際に表示させる文献データを件数文だけ limit 関数によって、取り出してくる。limit 関数の後に検索結果の何番目から取り出すか指定し、そこから表示件数分（pagesize）のデータだけ取り出す。例えば、検索結果の件数が 35 件で、表示件数が 20 件の場合ＳＱＬ文に「limit ０, ２０」が追加され、まず０番目から２０件分のデータが取り出される。（資料編、zogai_result.php、１１７行目参照）残りの１５件は、次の１５件というリンクが表示され、リンクをクリックした際に、ＧＥＴメソッドでデータが次のページに送られ、計算を行う。ここでは、２１番目から２０件分取り出せという指示になる。しかし、残り件数が１５件のため、実際は１５件分だけ取り出される。

２章の改善点より最後のページの表示件数と、リンクの表示数が合わない問題について、以下のように改良している。ページ番号は、page 変数で管理しており、次のページに進むと page 変数が＋１され、戻ると-１される。if 文を使い、全体のページ数－現在のページ＝１になる場合のみ、最後のページに表示される件数を計算してリンク表示に書きだすようにしている。例えば、全体のページが６ページだとして、最後のページの一つ前のページ、つまり５ページを表示する際、計算結果が１になるので、残りの件数を計算してリンクを作成している。（資料編、zogai_result.php、ソースコード１２０～１３０行目参照）

以上のように検索結果が、指定した表示件数を超える場合は、ページ送りシステムにより複数のページに分割されて表示される。これによ

り、素早く検索処理、表示を行うことが可能である。

（iii）エラー処理

何も入力せずに search ボタンを押してしまった場合、検索語句を入力するように警告するエラーを返している。また、検索語句が１文字だった場合は、２文字以上入力して検索させるようにエラーを返すことにしている。これは、１文字だけの項目検索では膨大なデータが呼び出され、検索者の意図するデータを探すのに手間がかかると判断したためと、チベット文字は、基本的に複数の文字から成り立つという点からである。具体的には、mb_strlen 関数で受け取った文字列の長さを測り、1文字だった場合はエラー表示を行うようにしている。その際、受け取るデータの文字エンコーディングは、チベット文字が shift_jis や euc-jp に対応していないため utf-8 を指定している。（資料編、zogai_result.php、２９～３６行目参照）

（iv）検索結果の表示

検索結果は、一件のデータずつ、テーブルで並ぶようにしている。（資料編、ソースコード２００番～２０８番、zogai_result.php 参照）また、タイトル、著者名は他に比べて見やすいようにＣＳＳファイルで font-size を定義し、大きく表示するように指定している。そして、チベット文字は、縦にも連結して伸びるため、抜き出したデータが２行以上にわたる場合、下の行の文字が潰れてしまう不具合が発生した。そのため、ＣＳＳファイルで line-height を定義し、行間を多めにとることで対処している。

また、序論で述べたように、検索した語句は赤色で表示するようにしており、検索語句がどの部分にヒットしているのか一目で判断できる。その処理は以下の通りである。まず、implode 関数で検索語句を|で区

切り（）でひとまとめにし、各項目の re_pieces 変数に代入する。これにより、（）の中の部分は、検索語句だということを定義する。そして、表示する際に preg_replace で検索結果から抜き出したデータの中から re_pieces 変数（検索語句）だけを赤く表示するように指示している。赤く表示させるのは、CSS で表示色を定義し、（資料編、zogai.css、１２～１４行目参照）データから検索語句を抜き出すのは PHP で処理している。（資料編、zogai_result.php、４１行目、５５行目、１９２～１９８行目参照）ただし、テキストナンバー、key_letter 一覧検索では、複雑な検索語句を入力しないので表示色の変更は行っていない。

（v）再検索処理

検索する際にトップ画面に戻る手間を省くため、検索結果を表示する上部に、検索を行える入力フォームを配置した。入力フォームには、トップ画面で検索した語句が入るようになっている。検索した語句が何であるか忘れないためという意味と絞り込み検索する際に、追加の語句を入力するだけで良いように手間を省く意味を込めた。検索結果画面の再検索フォームは、絞り込み検索のこともあるので意図的にクリアボタンは作成しておらず、リサーチボタンのみにしている。また、結果表示後であっても表示件数を変更できるようにしている。（資料編、zogai_result.php、１４３～１６８行目参照）

（5）インターネット・サーバーにアップロードする

（i）使用するサーバー

ローカルホストで異常なく検索システムが動作するのを確認した後、実際にインターネットのサーバーに制作したファイルをアップロードするに至った。アップロードについて福田先生と相談した結果、卒業

制作物を今後、大谷大学で公開する可能性があるということで、レンタルサーバーではなく、大谷大学のサーバーにアップロードしようということになった。そのために、アップロードしたファイルを保存する場所を作成しなければならないのだが、いくら本学の学生であると言っても私には、大谷大学のサーバーを操作できる権限が与えられていない。そのため、教育研究支援課の卯川さんに御協力頂いた。そして、大谷大学の www2.otani.ac.jp/fkdsemi/の中に私のフォルダと卒業制作物用の tibetan フォルダを作成して頂いた。最終的に制作物の場所は、http://www2.otani.ac.jp//fkdsemi/0548038xz/tibetan/にアップロードし保存している。

（ii）意図しないエラー

インターネット・サーバーにファイルをアップロードして動作確認してみると、検索結果画面で、入力していたチベット文字が正しく表示されない「文字化け現象」を起こし、データベースからデータを抜き出せなかった。ローカルホストとインターネットのサーバーでは、環境が違うので、意図しないエラーが発生してしまったのである。

まず、表示している検索語句が化けているとなるとデータベースにアクセスする際の文字も化けていると考え、文字化けの改善に取り掛かった。文字化けの主な原因は、文字コードを間違ったエンコーディングで表示しようとした際に起こる場合と、サーバーが指定した文字コードに対応していない場合である。まず、インターネット・サーバーが今回使用している文字コード utf-8 に対応しているか、卯川さんに尋ねたところ問題ないとの返答だったので、文字コードをエンコードする際に不具合が発生していると判断した。

そこで、データの受け取りを行っている PHP の部分に文字エンコー

ドの設定を行った。mb_language 関数を使い、使用する言語を uni（utf-8）に指定し、mb_internal_encoding 関数で内部文字エンコーディングを utf-8 に指定した。そして、mb_http_input で入力された文字エンコーディングを検出し、mb_http_output 関数で出力する際に utf-8 になるように設定した。そして、動作確認してみたが、入力したチベット文字は、文字化けを起こしたままであった。そのため、この箇所を//でコメントアウトして機能しないようにしている。（資料編、zogai_result.php、９～１２行目参照）

ローカルホスト内では、うまく動作していたので、インターネット・サーバー側に何か問題がないか、卯川さんに現状を報告するとともに確認して頂いた。すると、クライアント側（私側）で文字コードを設定しても管理者側（インターネット・サーバー）の設定が優先されているかもしれないとのご意見を頂き、管理者側の設定を変更して頂けることとなった。その結果、入力したチベット文字が文字化けを起こさないように改善された。以下、卯川さんから伺ったことである。

管理者側の PHP の設定ファイル PHP.ini を開き、文字コードの設定を確認したところ、default_charset が「Shift_JIS」になっていたので、utf-8 に変更したとのことである。

文字化けは改善されたが、検索したデータの取り出しがうまくいかないエラーは改善されなかった。データベースにアクセスする sql 文を確認したところ、入力されている検索語句は、チベット文字で表示されていた。PHP でのデータの受け渡しは正確に行われているとすると、MySQL 側に問題があると考えた。コマンドプロンプトで、直接登録されているデータベースの中身を確認したところ、チベット文字が文字化けを起こした状態で登録されていた。原因は、インターネット・サー

バーで文献データの sql ファイルをデータベースに読み込ませる際に文字コードを指定していなかったためであった。すぐさま、データベースを一度削除し、新たに、default-character-set で文字コードを utf-8 に指定し、sql ファイルを読み込むように命令した。また、念のためデータベースを定義する sql ファイルも utf-8 で読み込ませている。コマンドプロンプトでデータベースを確認すると、今回は、文字化けせずにデータを登録できていたので再度、動作確認を行った。しかし、データベース、サーバーの設定はともに utf-8 に登録されているにも関わらずデータをデータベースから抜き出せないエラーが発生してしまった。

完全に手詰まりに陥ってしまう前に、一度インターネットで同じようにデータベースに utf-8 を使用した際にエラーの事例がないか検索を行った。そのところ、多くの方が同じようにデータベースのエラーに困っておられ、改善策も示されていた。そこで、HP(4)を参考に sql 文を実行する際にクライアント側の使用する文字コードを MySQL に通知する方法を試してみた。データベースにアクセスする際に「set names utf8」と定義を行うと、見事データベースからデータを取り出せるように改善できたのである。（資料編、zogai_result.php、１３行目参照）

以上の工程を経て、検索システムはインターネット・サーバーで稼働するに至った。

(6) 動作テストの結果

インターネット・サーバーで蔵外検索が問題なく動作できるようになったので現サイトに修正点がないか本学の三宅先生に動作テストをして頂いた。三宅先生は、チベット文献の研究者であり、新北京版の制作にも携わっておられるので、チベット文字を用いた検索システム構築に

有益な情報が手に入ると考えたためである。そして、頂いた意見は以下の通りである。

(i) 一覧表示の削除

key_letter 一覧表示は、チベット文字の基字を表にしたものから、(資料、図３参照) クリックした基字のデータが一覧で表示されるようになっている。しかし、基字から抜き出したデータ一覧を一つずつ閲覧して、目的のデータを探すのは時間がかかり、それなら検索語句を入力したほうが容易に必要なデータを取り出せるという意見を頂いた。さらに、ソートのデータが作られていないため、基字の後に続く文字がばらばらになっているという意見も頂いた。

ソートというのは、例えば日本語で言うと、「あ」の後に続く語が、「あい、あう、あえ、あお」に並ぶようにデータの集合を一定の規則に従って並べることである。しかし、今回の key_letter 一覧表示に関して、ソートのためのデータは、作られておらず、検索結果を表示する際に、「あえ、あな、あは、」のように後に続く文字が無茶苦茶にデータが抜き出されるため、目的のデータを探すのに非常に時間がかかる。そのため、項目別に入力検索ができる現在では、必要ないであろうと判断したので一覧表示検索を削除するに至った。

(ii) 全データが表示される場合

チベット文字では、音節の切れ目ごとに書き入れるツェクという点がある。入力フォームにツェク、スペース、ツェクと入力して検索すると全てのデータが表示される。これは、チベット文字は音節の切れ目に、必ずツェクが入ることと、検索処理の際にツェク、スペース、ツェクで２文字以上だと判断して、エラーを返さずに検索しているためである。そのため、ツェクが含まれるデータを検索するので、当然全てのデータ

が表示される。その場合、抜き出したデータをコピーして、テキストメモなどにペーストすることで簡単に全データを抜きとれてしまう。トップ画面に Copyright: (c) と表示し、警告はしているものの、データ作成者からすれば気持ちの良いものではない。そのため、検索結果が５００件以上の場合は、検索語句を絞るようにエラーを返して、全データを表示しないように処置している。

（iii）入力フォームについて

入力フォームにおいて、チベット文字が結合した際、下の部分が切れて表示されないという不具合が発見された。入力フォームを小さく文字のサイズを大きくしたためだと考え、入力フォームを大きくしてみたが、改善できなかった。次に CSS ファイルで padding を定義して文字の周りに空白を作ったが、結局は、一部の文字が切れてしまう不具合は、改善できなかった。理由がわからず、試行錯誤した結果、文字を入力するフォームがチベット文字を理解していないのでは、と思いフォント設定を行った。CSS ファイルで font-family を定義し、この入力フォームには、チベット文字が入ると定義したところ文字が切れる不具合を改善できた。

（iv）表示の違い

検索結果の再検索フォームに表示されるチベット文字と検索結果で表示されるチベット文字のフォントが違うという意見を頂いた。チベット文字のフォントは、標準なら windows(vista) では Microsoft Himalaya が一種類だけ入っているが、Mac(MacOS X Leopard) には kailasa と kokonor の２種類のフォントが入っている。活字風の kailasa と木版風の kokonor では、表示が変わるのでどちらか一つにまとめたほうが見栄えが良いだろうという意見をもとに表示するフォントを限定すること

とした。上記の入力フォームについて、で述べたように CSS ファイルで font-family を定義し、表示するフォントを入力フォーム、結果表示部、双方とも kailasa に限定している。

4　結論

(1) 残された問題

(i) OS による表示問題

使用している OS によって標準で搭載されているチベット文字を表示する環境に違いがある。Mac(MacOS X Leopard) には 2 種類のフォントが、windows(vista) には 1 種類のフォントが搭載されている。OS によって搭載されているフォントが違うため、本サイトを winodows で表示した際に搭載フォントの違いで Mac と比べると文字が細く小さく表示されてしまう。(資料編、図 2 － 2 参照) 私は、常に動作確認を Mac で行っていたこともあり、Mac で表示されることを前提に考えていたので windows で表示された場合どう表示されるかまで考えていなかった。この問題は、海外のチベット文字を使った検索サイトにも当てはまり、どちらかの OS に合わせて表示が変わるので、逆に Mac で見づらいということもある。この問題を、どの OS で表示させても同じように表示できないかと三宅先生にご指摘頂いた。

ご指摘頂いた後、どのようにプログラミングを組めば、どの OS でも同じように表示されるか自分なりに考えてみた。携帯電話は、様々な機種が出ているが見ているサイトはどの機種でも同じように、又は機種毎に見易いように表示されている。このことを参考に、ブラウザで表示される際に使用している OS を判断し、その OS に合った表示サイズに変換する機能を作成すれば可能ではないだろうかと考えた。JavaScript

で、ブラウザの種類や OS を認識する機能を見つけたが、処理が複雑化し、意図しないエラーが発生するのを避けるため、実現するには至らなかった。

（ii）データの不備

検索システムを動作テストしていくなかで、多数の誤字、脱字が発見された。人間が、データを作成しているので、誤字、脱字は避けられないが３５４１件もデータがあるうえ、間違いがあるデータの確認にも時間がかかる。データベース化して容易に発見できるようになったとはいえ、誤字・脱字が残っている可能性は捨て切れない。私は、チベット文字について全くと言っていいほど知識がないので、どの部分が間違っているか、気づけなかった。そのため、検索システムを使って発見されたデータの不備は、専門家の方々に修正をお願いしなければならないのが現状である。以上のことから、完璧に誤字、脱字がない文献データのデータベース化には至らなかった。

（iii）unicode による問題

preg_replace 関数を使い、抜き出したデータの中から検索語句だけを赤く表示する件について、三宅先生から指摘して頂いたことがある。それは、チベット文字の結合文字をわざと結合しない状態で検索を行った場合、検索結果も結合しない状態で結合文字が分離して赤く表示されることである。（資料編、図１１～１４参照）これは、検索語句だけを他の抜き出したデータから分けているためだと思われるが、その他に unicode の原理的な問題も含まれる。unicode について、以下のように福田先生から伺った。

unicode では、チベット文字のような複雑な文字は、前後の文字データの中で特定の配列になったときに、特定の複合文字を表示するよう

に作られている。したがって、文字と文字との間にスペースを入れたり、また複合文字の一部だけを強調表示する html タグを挿入してしまうと、前後の文字との関係がなくなってしまって、文字が結合して表示されなくなってしまう。

理想は、結合文字を結合しない状態で検索しても検索結果は、結合した状態で表示し、なおかつ検索語句だけ表示色が違うことであるが、この問題を改善するためには、チベット文字を理解し、結合しない状態で検索された場合は、その文字に結合する文字を繋げて表示するようにプログラミングを行えば回避できると思う。しかし、この問題は、unicode の原理に関わる問題なので対処できなかった。結合文字を結合しないまま検索されないことを願うしかないのである。

（iv）ソートデータ

初めて目録検索を見かけた方が、この目録検索には、どのようなデータが収録されているのか、一覧表示して確認したいのではないだろうかという意見を福田先生から頂いている。そのためには、見やすい表示に並ぶようにソートのデータを作成する必要がある。しかし、基になった蔵外文献目録のテキストデータには、ソートに関するデータは、登録されていない。３５４１件のデータにソートデータを追加作成するのは非常に大変であり、チベット文字の知識も必要になる。そのため、今回は一覧表示検索を削除して見送ることととした。ソートデータについては今後、研究者の方がデータを追加するか否か、判断をお任せするしかなかった。

(2) インターネット・サーバーで unicode を使用するために

今回の経験から、チベット文字のように utf-8 でしか文字を表示できない場合、インターネット・サーバーでの運用は以下の通りである。

1. 管理者にサーバーの文字エンコードが utf-8 に対応しているか確認し、文字コードを utf-8 に変更して頂けるか確認する。そして、管理者側の PHP の設定ファイル PHP.ini を開き、文字コードを指定している default-charaset を探し utf-8 に変更して頂く。
2. データベースを定義する sql ファイル、データベースに挿入する insert 文が入ったデータの sql ファイルをコマンドプロンプトで読み込ませる際は、default-character-set で文字コードを指定して読み込ませる。
   例「mysql -u tibet –default-character-set=”utf8” < zogai.sql」
3. データベースを管理する MySQL ソフトに対し sql 文の文字コードの指定を行う。データベースに接続する際に「SET NAMES utf8」と定義する。(資料編、zogai_result.php、１３行目参照)

以上のように、文字コードの指定を全てのファイル、使用するプログラミング言語に定義すればエラーを回避できる。

(3) 終わりに

今回の制作にあたり、テーマに掲げた「蔵外文献目録のオンラインデータベース化について」は達成できたと思える。制作した蔵外検索システムは、インターネット・サーバー上で異常なく動作しており、今後のサイトの在りようについては、大谷大学の研究者の方にお任せする。

私は、今回の検索システムを制作する上で、常に気をつけた点がある。それは、検索システムが簡潔にまとめられているかという点である。イ

ンターネット・サーバーにアップロードするということは、世界中の人から見られるということである。そのため、一目見て、わかりやすいシステムになっているか、使い勝手は良いのかといったことを常に念頭に置き、尽力したつもりである。人によっては、検索システムがあまりに質素と感じられる方もおられるだろうが、極力無駄な機能は省きたいと考えた結果が今回の蔵外検索に表れていると思う。

制作を通して、学んだ知識は本論中に記してあるので、以後のチベット文字を使ったデータベース作成に多いに役立てて欲しいと思う。そして、少しでもチベット文化の発展に繋がることを願ってやまない。

注

（1）『Peking Tripitaka Online Search』
http://web.otani.ac.jp/cri/twrp/tibdate/Peking_online_search.html

（2）新『Peking Tripitaka Online Search』
http://www2.otani.ac.jp//fkdsemi/0548038xz/tibetan/search.PHP

（3）URL の後ろに「？変数名＝値」を追記することで、その変数名と値を URL に送信する。

（4）WindowsVista+Apache2.0+PHP5+MySQL5.0 による Web アプリケーション
http://www.yamada-lab.org/doc/win/MySQL5/index.html

文献表

PHP Manual
http://www.PHP.net/manual/ja/

MySQL 5.1 リファレンスマニュアル
http://dev.MySQL.com/doc/refman/5.1/ja/index.html

石田豊
2005 『MySQL 入門以前』毎日コミュニケーションズ