unicode を利用したチベット語文献検索システムについて

美濃部　春菜

目　次

――unicode を利用したチベット語文献検索システムについて――

1　序論

(1) 経緯

本学図書館には 1330 函にわたる「北京版西蔵大蔵経」が所蔵されている。この「北京版西蔵大蔵経」は中国以外では本学図書館とフランスの国立図書館にしか置かれていない。そのため、「北京版西蔵大蔵経」を必要としている人は、本学かフランスに赴く他に方法がない。しかし、それは非常に不便である。これを解消するため、「北京版西蔵大蔵経」は 1955 年～61 年に鈴木学術財団から影印刊行され、世界中で容易に閲覧することが可能になった。

「北京版西蔵大蔵経」が影印出版されたのと同時に目録・索引も刊行された。しかし、1330 函にわたる「北京版西蔵大蔵経」は目録・索引自体もかなり膨大な量である。その中から、必要とする文献情報を探し出すのは容易なことではない。また、曖昧な情報（タイトルの一部分だけを知っているなど）だけでは必要とする文献情報を手に入れることは非常に困難である。

そこで 2005 年、「北京版西蔵大蔵経」をさらに便利で使いやすくするために、真宗総合研究所により「北京版西蔵大蔵経」をオンライン化するに至った(1)。これにより、曖昧な情報での検索や、著者名からの検索が可能になり、実際に冊子を手にしなくても目録検索を行うことが可能になった。

(2) 現状と問題点

（１）で述べた経緯により、北京版チベット文献検索サイトは制作された（以下、「旧サイト」と呼ぶ)。しかし、完成当初はパソコンの OS

の問題により、本来チベット文字であるチベット文献をローマ字表記でしか検索・表示することが出来なかった。チベット文献を検索するにあたって、一番大きな問題ではあるが、他にもいくつか不都合な点が見つかった。

旧サイトでは、フリーワードによる検索を主としている。フォームに調べたい単語を入力し、「search」ボタンを押して検索を行う。一つのテキストボックスには一つなぎの単語（「zla ba'i」など）を入力できるが、違う項目の単語や、離れている単語を検索する場合は、二つ目のテキストボックスに入力しなければならない (資料編、図 1 参照)　。

また、一つ目のテキストボックスと二つ目のテキストボックスについて and 検索で検索を行うか、or 検索で検索を行うかをラジオボタンで選択することが出来る。しかし検索は、データを単純に全文検索して行っている。そのため、検索者が意図していないデータまでヒットする可能性がある。資料編の図 2 を見ていただければ分かるが、1 件目の文献データでは 2 つの検索語が、共にタイトルの該当箇所にヒットして結果として表示されている。しかし次の文献データでは、一つはタイトルに、一つは人名にヒットしている。このように、文献データによって検索されている項目が違うのは、検索者の意図を無視した結果である。検索する際の精度が低くなることは非常に問題である。

表示についても同様に、ヒットしたデータについて単純に並べて表示しているだけである。検索語句によっては 100 件以上もヒットするが、検索結果の画面ではそれがページ送りもせずに表示されてしまう（資料編、図 3 参照)。精度も低いので、必要な情報を手に入れるまで非常に時間がかかってしまうことになる。

以上の観点から、旧サイトは複数の問題を抱えており、新しくサイト

を作るべきである。

(3) 目標

新しい北京版チベット文献検索サイト（以下、「新サイト」と呼ぶ）をつくるにあたり、まず第一にチベット文字による検索・表示が行えることが必要である。本来、文献はチベット文字で表記されている。そのため、検索結果でもチベット文字で表示されるのが好ましい。また、ローマ字だけで表示されている状況は非常に見づらい。チベット仏教の研究をしておられる福田先生にチベット文字に関して聞いたところ、チベット文字は非常に複雑な構造を持っている。チベット文字は横に並ぶだけでなく、縦にも結合し、結合するときには形も変わる。そのように縦横に結合して、最大 7 文字が合わさって一音節になる。一つの単語は 2 音節以上からなり、さらに複数の単語が集まって一つの仏教用語となる。ローマ字で表記されていただけでは、一見して一音節の構造が把握しにくく、複数の音節が連なった単語に至っては、最初から全てを読んでチベット文字に直していかなければ理解できない。意味を理解するのも時間がかかるため、この作業は非常に面倒なものである。以上により、一音節の構造が一目で理解できるチベット文字で表示されるのが好ましいのである。

その他にも改善点がいくつかあると思われるため、三宅先生に伺った。三宅先生は、旧サイトのデータを作成されており、また実際に旧サイトを利用しておられた。つまり、制作者であり、利用者でもある方である。三宅先生に伺うことで、利用者の観点だけでなく、制作者の観点からもご意見をいただくことが出来ると考えた。そして旧サイトよりもさらに特化したサイト作りができ、便利で利用価値のある制作物が完成

されるのではないかと考えたからである。実際、私自身だけでは考え付かなかった機能をいくつかあげていただいた。その結果、以下の機能を追加および旧サイトから継続することが決定した。

その機能は、検索方法、表示方法の二点である。

検索方法については、出来るだけ高い精度の検索が行えることに尽きる。タイトルから特定の文献を調べたいのにもかかわらず、著者名に同じ検索語句があるからヒットするというのは非常に厄介である。著者名よりヒットした検索結果は、検索者の意図した結果ではないため、すべて無駄な結果である。このような、無駄な検索を行わないために、タイトルで検索した場合はタイトルでの検索結果のみを、著者名で検索した場合は著者名での検索結果のみを表示するように改善しなければならない。

また、表示方法に関しては、見易さ、必要とする情報を手早く発見できるようにする。旧サイトにも「検索結果中に検索した語句があれば赤色で表示する」という機能があるが、これはどの部分が検索されているのか容易にわかるため、新サイトでも利用するべきである。しかし、検索結果が何十件もあるにもかかわらず、一つのページに表示されているのは非常に見づらいため、一定の件数以上が表示されれば次のページにするという工夫が必要である。

以上の点に留意し、北京版チベット文献検索サイトを制作する。

2　制作するにあたって

(1) 利用者の限定

北京版チベット文献検索サイトは、利用者が限定される。すなわち、チベット学研究者またはチベット文献を必要としている方々である。それゆえ、機能や表示に関して上記の方々が利用しやすいものを目指す。

(2) チベット文字で検索・表示できる環境

チベット文字を表示するために、私は文字コードとして UTF-8 を採用した。チベット文字は、以前は本学のサイトで配布されていた。しかし、それは OS 本体とは別に配布されていたため、普及度が今ひとつであった。それが、2007 年 10 月 26 日に発売された MacOS　の Leopard に搭載されることになり、全てのユーザが最初から利用できる環境となった。

Windows の場合は、UTF-8 は XP でも採用されていたが、unicode のチベット文字環境が提供されるようになったのは Vista からである。

Macintosh であっても Windows であってもチベット文字の普及はこれからの段階である。しかし、チベット文字の表示・入力が可能になった段階で、チベット文字でのデータを提供する必要が本学にはある。同時に、従来から利用しているユーザに対してのサポートも続ける必要がある。そのため、ローマ字での検索・表示も行えるようにする。

(3) 使用した技術

旧サイトでは、チベット文献のデータはすべてテキストデータであった（資料編、図 4 参照)。しかし、このままでは実際に検索を行う際に不備がでてしまう他、データの追記・修正に当たる場合に非常に不便で

ある。よって、新サイトではテキストデータのデータベース化を行うことにした。そのためにデータベースを管理するソフトとして、MySQL を利用した。

また、検索システムには PHP 言語を利用し、デザインやその他サイト全体において HTML 言語、CSS 言語を利用した。

3　本論

(1) データベース化

(i) データベース化の必要性

まず、はじめに取り掛からなければならなかったのは、文献データのデータベース化である。データベース化にはいくつかの利点がある。

１つ目は、データの把握が容易になることである。テキストデータのままでは、どこに何のデータがかかれているのか一見しただけでは判別できない（資料編、図 4 参照)。旧サイトで利用されているテキストデータは文献のデータごとに<dt>でわけ、タイトル、北京版、その他ごとに<dd>で区切る非常に単純な構造である。「北京版西蔵大蔵経」の場合、データ件数は 5000 件以上にのぼるため、効率よくデータベース化し、項目やデータの内容などをすぐに判別できる方が適当である。データベース化することで、データを容易に把握することが出来、データの不備や誤字を見つけることが可能になる。

２つ目は、データの変更、追加が容易になることである。1 つ目と重なるところもあるが、5000 件以上のデータをテキストデータとして羅列していては、どこに何が書かれているのか把握できない。そのため、もし、データの確認を行おうとすれば、一つずつ吟味していくしか方法

がない。しかし、データベースの場合、項目ごとにはっきりと分かれて表示できるので、どの項目に何が書かれているのかすぐに把握でき、変更もすぐに行える。追加に関しても同じことが言える。文献データに対して何か新しい項目を付け足す場合、テキストデータでは各文献ごとに新しい項目を毎回追記していかなければならない。これは非常に労力がかかり、入力時のミスも増える。データベースであれば、新しい項目を追加することも簡単に行え、各データの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も遥かに短時間で項目の追加を行うことが可能である。

3 つ目は、検索の複雑化に対応できることである。旧サイトは全文検索を採用し、すべてのデータの中から検索語にヒットするものを結果として表示していた。しかし全文検索ゆえ、利用者が意図しない結果が表示される可能性があり、精度の低いものにならざるを得なかった。データベース化を行えばデータを項目別に分けることが出来るため、利用者が検索したい項目を選び、その項目内で検索したい語句を入力することで、確実に利用者が要求する結果を導き出せる。精度も旧サイトより遥かに高くなると考える。

以上、3 点の利点により、文献データはテキストデータではなくデータベース化する必要があると考える。

（ii）項目の追加

文献データは既にローマ字で表記された、チベット語タイトル、サンスクリット語タイトル、著者（チベット語/サンスクリット語）、翻訳者、校訂者、北京版 No.、デルゲ版 No.、ナルタン版、金写の 10 個の項目があった。この項目は新サイトを作るうえで最低限の項目であるため、そのまま採用した。さらに、タイトルにはチベット文字とサンスクリット

語転写文字の項目をそれぞれ追加し、人名も同様に項目を追加した。ただし、著者、翻訳者、校訂者は制作の都合により 1 つの項目とした。また個々のデータを個別に認識できるために、id という項目を追加した。

（iii）データベース化の過程

データベース化するにあたって、一番初めに行うことはテキストデータの把握であった。テキストデータは先述したとおり、<dt>及び<dd>により項目が分けられていたが、非常に単純な分け方であった。このままデータベース化を行っても、それぞれの項目をうまくわけることは出来ない。確実に項目化するために独自にプログラムを作り、より細かく区別するようにした（資料編、資料１参照)。

そうして、項目をタブで区切った後、エクセルにタブ区切りファイルとしてテキストデータを読み込ませた。しかし、それだけではデータベースとして利用することは不可能であった。それは、テキストデータの段階で既に起こっていたデータの不備によるものであった。例えば、データのない項目を無視して作っていたため、他のデータが違う項目に入っていたり、明らかな誤字が多数あったりしたからである。tab.py を使うことにより、手早く項目別に分けることが出来たが、最終的には一つずつ正しいデータが入力されているか確認し、修正せざるを得なかった。

このように、一通りチェックしたデータにチベット文字とサンスクリット語転写文字の項目の追加を行った。ローマ字で表記されたチベット語やサンスクリット語はプログラミングでチベット文字やサンスクリット語転写文字に直した。しかし、著者・翻訳者・校訂者の項目はチベット語とサンスクリット語が混在している。そのため、チベット語の部分はチベット文字に、サンスクリット語の部分はサンスクリット語転

写文字に直す必要があった。そこで、的確に変換するためにそれぞれの言語の特性を生かした。チベット語は文の区切りに”\” が使用されている。サンスクリット語は文の区切りに”.” が使用されている。これを生かしてプログラム内でチベット語とサンスクリット語を識別し、チベット語とサンスクリット語が混在したデータでも的確に変換することが出来た（資料編、図 5 参照)。

そうして、タイトル及び著者・翻訳者・校訂者のチベット文字とサンスクリット語転写文字の項目を追加したエクセルファイルをデータベースに読み込ませた。まず、エクセルのファイルをタブ区切りファイルとしてエディタで読み込んだ。その後、正規表現で、文頭に「values (”」を追加し、各タブごとに「”,”」、文末に「”);」を追加する。最後に、データベースの作成のコマンド、SQL 文を追記し、文献データのデータベース化が完了した（資料編、図 6 参照)。

(2) UTF-8 の使用

チベット文字を表示するために、UTF-8 を使用した。これは、PHP や MySQL、ブラウザなど全てで UTF-8 を用いなければ表示することが出来ないためである。しかし、新サイトではデータベース、PHP ファイル間でデータの受け渡しが行われるため、どのように受け渡すかが重要であった。

まず、データベースはもともとチベット文字やサンスクリット語転写文字を含まれている。そのため、データベース内で文字化けを防ぐために、データベース自体を UTF-8 で保存することにした。次に、保存したデータベースを PHP 言語、MySQL を利用してデータを取り出してみた。すると、予想通り結果画面で文字化けしてしまった。しかし、ブ

ラウザ上でエンコードを「UTF-8」にしてみたところ、データベースに入力されているとおりのデータが表示された。ただし、PHP のファイル内に直接書いた文字（見出し語）などが逆に文字化けしてしまう、という事態に陥った。

取り出したデータの文字化け、PHP ファイル内に直接書いた文字列の文字化けを改善するために、まずブラウザ上でエンコードを変更する手間を省くため、事前に HTML 言語内の meta タグの中に charset として UTF-8 を指定した。また、PHP ファイル内に直接書いた文字が文字化けするのを避けるために、それらをすべて変数に置き換え、mb_convert_encoding という関数により文字列も UTF-8 化した。このようにすることにより、データベースから取り出したデータも、PHP ファイル内に直接書き込んだ文字列も UTF-8 というエンコードで正しく表示することが可能になった。

次に、別のページで入力された文字列が UTF-8 であった場合、どのようにして正しく取り出してくるかについて考えた。まず、mb_language 関数でファイルの元々の言語を UTF-8 に設定し、mb_http_input 関数で入力された文字列のエンコードを検出、mb_http_output 関数で出力する際に UTF-8 になるように設定し、上記のすべての関数を、文字を入力させているページ及び、PHP 言語によって実際にデータベースからデータを取り出すページに使用してみた。しかし、入力された文字列をうまく、正確に取り出してくることは出来なかった。

そこで、新たな案として、ファイル全体のエンコードを元から UTF-8 にしてみることにした。関数で一つ一つ確実にエンコードの確認や、変換を行うことも有効な手段だと思ったが、うまくいかなかった。ファイル全体を UTF-8 にしてしまうことで、そのような関数を書く手間も省

け、また、エンコードが安定することで今後様々な機能を追加していってもトラブル回避につながると考えたためである。そうして、ファイル全体を UTF-8 にしてしまうことで、フォームに入力された文字列及び、その文字列を別ページに受け渡す際も、検索語として使用し、結果として出てきたチベット文字を表示する際も正確に表示されるようになった。

ゆえに、すべてのファイルのエンコードは UTF-8 を使用することにした。エンコードを UTF-8 にしたため、mb_convert_encoding 関数等がなくても正しく表示することが出来た。そのためこれらの関数は使用しなかった。ただし、ブラウザの状態も常に UTF-8 である必要があるため、charset の UTF-8 は引き続き採用した。

(3) サイトの構成

新しい北京版西蔵大蔵経目録検索サイトは、ほとんどを旧サイトと同じ構成で作成した。トップページは目録検索に関する説明と、実際に検索するための入力フォームがある（資料編、資料 3 参照)。実際に検索を行ったり、その他機能は一括して一つのページにまとめた（資料編、資料 4 参照)。ただし、後述する「ヴォリューム検索」は検索方法が異なっているので別のページとした（資料編、資料 5 参照)。また、どのページからも検索が行えるように、上部に検索フォームを設けた。その他に、文献データが含まれる SQL ファイルと、レイアウトのための CSS ファイルを全てのページに適用させた（資料編、資料 2 参照)。

（4） 検索方式

（i） 項目別検索

データベースのデータを検索する際、プルダウンメニューから検索したい項目を選び、下部のフォームに項目に相応する語句を入力して検索するようにした（資料編、図 7 参照)。これは、既に存在する文献データから闇雲に検索するのではなく、ある程度目的を持って検索を行う専門家の意向に即した形にしたためである。一般には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に対して検索を行うものや、詳細検索などと呼ばれる項目間で論理演算子を用いることが出来る検索なども存在するが、今回は採用しなかった。表示に関しても非常にコンパクトに収めることができるため、項目別検索を採用した。また、北京版文献検索サイトということで、北京版や、デルゲ版から検索する可能性はあるが、ナルタン版や金写では検索する可能性が非常に少ないため、結果としては出力するが、検索項目にはしなかった。

（ii） and 検索

何か検索する場合、大抵の人は 1 つの単語だけではなく、より精度を高めるために複数の単語をつなげて検索するだろう。今回の場合も文献データが 5000 件以上にも及ぶため、1 つの単語だけでは実際に必要としているデータとは違うデータがいくつもヒットする可能性がある。そのような事態をさけるため、また、なるべく少ない操作で複数の単語で検索できるように、検索語句を半角スペースで区切ることで and 検索が行えるようにした。or 検索を使用しなかったのは、たくさんのデータをヒットさせるより、より的確で検索者がすばやく目的のデータにたどり着くことに焦点を置いたためである。精度を高めるということも 1 つの目標であったので、and 検索だけを採用することにした。

実際には、まずフォームに入力されたデータを変数に取り出し、explode 関数を使って半角スペースを区切りとしてとってきたデータのリスト化を行う（資料編、資料 4：47 行目参照)。そして、リスト化された検索語の単語数を調べ、その個数分、SQL の WHERE 句で条件として付与されるように繰り返し文で、SQL 文に挿入する（資料編、資料 4：93 行目～107 行目参照)。事前にヒットする件数だけを調べ、それを検索結果件数として出力を行う。

また、検索された結果が 100 件以上超えるときは、絞込み検索を行うようにエラー文を返すようにした。

（iii）数値による検索

北京版、デルゲ版はその文献データが北京版やデルゲ版のどの巻のどの段落に書かれているかがデータとして入力されている。検索者は大抵の場合その巻の数字（4 桁）で調べるため、北京版とデルゲ版においては数字による検索を行い、他のページ等のデータは検索されないようにした。

具体的には、入力された数字を変数で取り出し printf 関数の中で正規表現をつかって巻の部分だけ検索されるようにした。巻は北京版では「[P.No.]」の後、デルゲ版では「[D.No.]」の後に必ず入力されている。さらに、それは確実に 4 桁である（「333」などは「0333」となっている)。この特性を利用して、入力された数字に［P.No.］（または [D.No.]）を数字の前方に追加したものに置き換えて検索するようにした（資料編、資料４：40 行目～44 行目)。「%04d」はもし、4 桁未満の数字が入力された場合でも数字の前方に「0」をつけ、必ず 4 桁の数字に変換する処理である。これにより、データの後方に入力されているページ数に該当する数字があっても検索されず、正確な検索結果が行えるようになった。

(5) 表示方法

(i) ラジオボタンによる表示形式の変更

新サイトの最大の利点は、「チベット文字（及びサンスクリット語転写文字）の表示が可能」ということである。しかし、そのためにはチベット文字が正確にされる環境でないと意味がない。そして、もともと北京版西蔵大蔵経はチベット文字で表記されているため、チベット文字が表示できる環境の検索者はチベット文字で表示する文献データを利用するだろう。対して、チベット文字が表示できない環境の検索者は今までどおりローマ字で表示された文献データを利用すると思われる。つまり、環境の違いによってチベット文字で表示できたり、ローマ字で表示できたりする必要がある。また、チベット文字で表示できる環境であれば、ローマ字での表示は必要でないし、ローマ字で表示する環境ではチベット文字を表示すれば文字化けが起こってしまう。

この課題を克服する案として、表示形式をラジオボタンで変更できる機能をつけた（資料編、図 7 参照)。ラジオボタンには「Tibetan」と「Roman」の 2 項目をつくり、「Tibetan」が選択されれば、結果表示もチベット文字（及びサンスクリット語転写文字）が表示され、「Roman」が選択されれば、検索結果にはローマ字での文献データが表示されるようにした。また、項目の初期値は「Tibetan」が選択されているようにした（資料編、資料 3：41 行目参照)。これは、チベット文字が表示できる環境の検索者は、ローマ字表示よりも文献データの内容が把握しやすいチベット文字が表示される方を選ぶだろうし、北京版西蔵大蔵経を利用する専門家の人々は大抵の場合チベット文字を表示できる環境を備えているだろうという考えからである。

（ii）ページ処理

旧サイトは、検索結果をページ送りもせず表示する形式であった。この表示方法では、数十件も表示されてしまうと、必要な文献データを見つけるために何度もスクロールしなければならない。これは、非常に不便である。新サイトの検索方法（P.12、「and 検索」参照）でスクロールする手間や、精度を高める機能として検索結果が 100 件以上になる場合は、エラーになるようにした。しかし、さらにスクロールする手間を省くために 20 件ごとに次のページが作られるように新たな機能を追加した。

初めに、pagesize という変数に 20 という数値を決めておく（資料編、資料 4：58 行目参照）。pagesize が 1 ページに表示される文献データの件数である。次に、検索結果の件数だけを先に求める（資料編、資料 4：61 行目～77 行目参照）。結果の全体件数を把握し、それを pagesize で割ることで、全体のページ数が求められる。そのために先に結果の件数だけを求めている。また、検索結果が一桁の場合、「次のページ」というリンクが表示されないように link という変数に if 文で条件をつける際にも検索結果の件数を利用している。

リンク設定が終われば、実際の検索を行う。その際、limit によって、指定された件数分取り出してくる（資料編、資料 4：106 行目参照）。limit の後に検索結果の何番目から取ってくるか指定し、そこから pagesize 分（20 件）のデータだけ取り出す。例えば、検索結果の件数が 32 件ならまず、0 番目から 20 件分のデータが取り出される。合計の件数は 32 件なので、残りの 12 件は次のページに送られるため、リンク設定で「次のページ」という項目が表示される。「次のページ」をクリックすると、limit で pagesize(20 件) とページ数を掛けたものが計算

され、21 番目から 20 件分が取り出される（ただし、この場合は残り件数が 12 件なため、12 件分だけ取り出される。）。

このようにして、20 件ごとに新しいページをつくることによって、たくさんヒットしても、短いスクロールで目的の文献データを探せるようになる。また、データを取り出してくる段階で、必要なデータだけ取り出すため、検索処理に時間がかからなくなった。ページ処理は「次のページ」を表示するだけでなく、2 ページ目に移動したときは、1 ページ目が表示できるように「前のページ」という項目も設置し、何度も検索する手間も省くことが出来た。

（iii）北京版一覧表示

検索者が利用していて欲しかった機能の中に、「ヴォリューム検索」というのがあった。ヴォリューム検索というのは、タイトルや、著者名などで検索したあと、文献データの北京版のヴォリュームから、そのヴォリュームに載っている他の文献データの一覧が検索できる、というものである。ヴォリューム検索が出来ると、ある文献データを調べていて、その前後に載っている文献データにはどのようなものがあるのか、というときに一目で把握することが可能になる。

実際には主に検索語にヒットしたデータを取り出した後に作業を行っている（資料編、資料 4：284 行目～289 行目参照）。まず、北京版 No. のデータが書かれている項目を、preg_match 関数で正規表現を利用して該当するデータを取り出す作業を行っている。丸括弧を 3 つ使用し、最初の丸括弧ではカンマ以外の文字列が続き、カンマとスペースが来るまでを切り取っている。2 つ目の丸括弧は、チベット語が続きカンマとスペースが来て、ヴォリューム番号のチベット文字アルファベットのみが来る。この部分のデータが volume.php に渡され、volume.php で検

索文字列として扱われる。最後の丸括弧は 2 つ目の丸括弧以降のページ番号などの部分である。

preg_match 関数でこれらの 3 つの部分に切り分け、vol という変数にリストとしてぞれぞれを代入して置く。その後、変数 vol を利用して、volume.php を呼び出すリンクを組み立てた。その際、ページや表示に必要なデータを同じように渡すために、アドレス欄に挿入した。

volume.php では、preg_match 関数でとりだしたヴォリュームを検索語として検索する（資料編、資料 5：73 行目参照)。そうして、取り出したデータが北京版のヴォリューム一覧となる（資料編、図 10 参照)。

このようにして、検索語にヒットしたデータから北京版のヴォリュームを一覧で表示することが可能になった。

(iv) 検索された語句に色をつける

結果表示のページで、検索されている語句に色をつけるという機能は、旧サイトにもあった。この機能は、どの単語がどの項目のどの語句にヒットして検索されたのか容易に判別することができるため、非常に便利な機能である。そのため、新サイトでも利用することにした。

新サイトでは、まず、取り出したデータをすべて sprintf 関数でひとまとめにし、そのデータに preg_replace 関数で検索語と同じ単語には色をつけるタグを挿入する、というプログラムを行った。しかし、その方法では本来、検索には使用していない項目に該当の検索語があった場合にも色がついていることが判明した。これでは本当に正しく色がついているとは言えない。そのため、取り出したデータすべてを sprintf 関数でまとめてしまうのではなく、一つ一つ条件が当てはまるときだけ色をつけるように変更した。条件を満たしているか判断するために if 関数を用いた。データを取り出した後にどの項目で検索したかを if 関数

で尋ね、その条件に当てはまる項目だけに preg_replace 関数を使って色をつけた（資料編、資料 4：295 行目～305 行目参照）。そうすることで、ほかの項目に同じ語句が存在していても色はつかなくなり、実際に検索されたときにヒットした語句だけに色がつくようになった。チベット文字に関しても、チベット文字で検索した場合は色がつくようになった（資料編、図 8、図 9 参照）。

しかし、表示形式を「Tibetan」と「Roman」の 2 通りにしているため、もし、ローマ字で検索語を入力し、表示形式を「Tibetan」にした場合は、結果で表示される文字はチベット文字であるため、検索された語句と表示された語句が違うので、色はつかない。これは検索語が、チベット文字のどの部分に相応するのか関連付けを行うことが出来ないからである。サンスクリット語転写文字についても同様に、色をつけることは出来なかった。また、北京版、デルゲ版は番号検索であり、検索した場合必ず 1 件のデータしか表示されないため、番号に色をつけるということは行わなかった。

4　結論

(1) 達成できなかった部分

(i) データベースの正規化

データベースにおいて重要な点は、「データの冗長性をなくし、整合性を持たせる」ということである。冗長性とは、ある情報が必要最低限より数多く存在することである。つまり、複数の項目に同じデータが複数存在している状態のことを指す。このような状態であると、同じデータについて変更点が見つかった場合、一つ一つ手直ししていかなくては

ならなく、非常に時間の無駄であり、見落としなどによる変更のミスが発生する場合がある。同じように整合性がない場合、同じデータであるはずなのにあるデータは正しく、あるデータは正しくないと言ったデータのミスが発生し、検索者が正しく検索することが出来ない、混乱するなどと言ったことが予想される。

このような事態を防ぐために、正規化というものがある。正規化は、上記のような冗長性をなくし、整合性を持たせるために必要な方法である。具体的には、複数の項目に同じデータが含まれている場合、そのデータは別のテーブルにまとめ、重なって登録されているものは簡略化し、リレーションをつないでリンクさせることを言う。このようにすることで、実際に書かれているデータは１つになり、変更点がある場合、それだけを変更するだけでリレーションをつないでいるすべてのデータに反映することが可能になる。今回の文献データの場合も、著者、翻訳者、校訂者において、同じ人物が同じ事柄を担当していることが判明した。データとしても無駄が多いため、正規化をする必要があった。

しかし、正規化を行ううちに、2 つの問題が浮上した。まず一つ目は「文献と人物の関係」である。北京版西蔵大蔵経では、一人の人物がある文献では著者を担当し、ある文献では校訂者を担当している等、複数の役割を担っている場合がある。また、一つの文献に複数の校訂者が存在する場合もある。つまり、文献と人物の間には多対多の関係が結ばれていることになる。そのため、単純に文献データのテーブルと、人物のテーブルをつくり、各々を繋げるだけでは完全な正規化にならないことが判明した。

そこで私は、文献データのテーブルと人物のテーブルの他に、2 つを繋げる役割を果たすサブテーブルを作ることにした。文献データと人物

にはそれぞれ固有の id を振っておき、サブテーブルでどの文献のどの役割を誰が果たしているかを id で判別できるようにした。一つの文献に複数の人物が関与している場合は、その都度別のレコードとして id を振った。つまり、一つの文献について複数のレコードが存在するようになった。検索を行うプログラムでは、文献のタイトルで検索された場合は、まず文献データだけを取り出す。文献データの id からサブテーブルを介して人物の id を取り出し、人物の id を使って人物のテーブルから必要なデータを取り出すようにした。

以上のことを踏まえ、実際のデータを使って行う前に、同じ構造を持った縮小版のサンプルでうまくデータを取り出すことが出来るか試してみた。しかしここで 2 つ目の問題が起こった。それは「データの表示」である。

先述したように、文献データと人物はサブテーブルを利用して各々にアクセスし、取り出すようにした。取り出し方は問題なかったのだが、表示方法に問題が出てしまった。文献データについて検索が行われた場合、検索語に対してヒットするものが何件あるか繰り返し文を使って検索を行っていた。そのなかでヒットした文献データについて、今度はサブデーブルに繰り返しアクセスし、人物のデータを取り出すようにした。すると、結果表示をする画面で、一つの文献に複数の人物を表示させるはずが、すべてがばらばらになり、同じ文献データが何度も表示してしまうようになってしまった。そこで、サブテーブルには何度もアクセスするが文献データの表示は一度にするために、GROUP BY で文献データをまとめることにした。すると今度は、人物のデータも 1 件しか表示されなくなってしまった。他にも繰り返しの方法などをかえ、何度か試してみたのだが、やはりうまく表示させることは出来なかった。

以上により、データベースとしては非常に不完全ではあるが、正規化を行わずに今あるデータをそのまま利用して、データベース化するに至った。

（ii）データの不備

データベースは、正規化できなかった点の他に、データ自体の不備も複数あった。多くは旧サイトの時から存在しているもので、文字化けや、誤字、脱字等である。また、データベース化を行った時、意図した項目にうまくデータを当てはめることが出来ず、違う項目に入っているものもあった。データベース化の途中で、データの不備が多数あることに気づき、最終的にすべての文献データについて手作業で一件一件確認を行った。そして、項目のずれなどの単純な不備についてはその都度修正し、脱字や不明箇所はその欄の状況を書き出し、三宅先生に不明箇所として提出し、回答を頂き、修正した。

しかし、所詮は人間が行った手作業でしかないことは確かである。一件一件確認したデータであるが、実際に制作物を作成している途中でもいくつかの間違いを発見した。発見したものについてはその都度修正するようにしたが、5000 件以上にも及ぶデータであるため、まだ間違いが含まれているデータがある可能性は捨てきれない。また、私が確認しただけではわからないような単語の間違いなど、専門家の方々に実際に利用していただいて気づく間違いも多数あるだろうと考える。

このような事情により、時間がある限り修正を行ったが、完璧な文献データのデータベース化は行うことが出来なかった。

（iii）検索画面の表示切替

三宅先生にベータテストをして頂き、一番ご指摘いただいたのは「検索画面の見にくさ」であった。旧サイトでは、検索・表示共にローマ字

でしか行うことが出来なかった。しかし、新サイトでチベット文字での検索・表示が可能になったため、検索や表示方法に幅を持たせるために、ローマ字で検索してもチベット文字で表示できるように表示形式と検索項目を分けて設置した。

しかし、利用する場合はそれが逆に不都合であるということであった。チベット文字が表示できる環境の人は検索もチベット文字で行い、ローマ字でしか表示できない人はローマ字での検索を行う。つまり、チベット文字が表示できる環境の人には検索項目にローマ字で検索できる項目は必要ないのではないかというものであった。そして、いくつも検索項目が並んでいるのは非常に検索しにくいということであった。

そこで、表示形式あるいは検索項目を選択することでもう一方を制御できる案を提示していただいた。例えば、表示形式で「Tibetan」を選択した場合、検索項目はチベット文字で検索する項目を載せる。「Roman」を選択した場合は、検索項目はローマ字で検索する項目だけ出てくるように、表示形式を切り替えることによって検索項目を変えるというものである。

ご指摘いただいた後、どのようにすれば切り替えることが出来るのか自分なりに検討してみた。その結果、javascript で似たような機能を作ることができるのではないかと考えた。何点か似たような機能を利用しているサイトも見つけ、実際にソースを見てみたが、事前にいくつかのソースを用意するなどの準備が必要だったため、実現するのは難しく、達成できなかった。

（iv）完全一致検索

三宅先生にご指摘いただいた点が、もうひとつある。それは、「記号等による完全一致検索」というものである。というのも、チベット文字

では 1 語であるものが、ローマ字では 2 語で表現される語句が存在するからである。そのためチベット文字で検索する場合とローマ字で検索する場合では、検索結果で表示される文献や、件数に違いが現れてしまう。2 語で検索した方は、間にスペースを入れてしまうため、前の語句と後の語句を別々に捉えて検索するからである。同じ語句で調べているにもかかわらず、検索結果が異なってしまうのはあまり好ましくない。

そこで、どちらの言語で入力した際も同じ検索結果になるように、間にスペースが入っていても、1 語として扱うように検索させる機能を追加してみてはどうかというご指摘があった。離れて検索してほしくない語句を”（ダブルクォーテーション）で囲むというものである。”であれば、文献データで使用されていないので、検索に支障をきたすことはない。これで、間にスペースが含まれていても 1 語として検索が可能になるというものである。

しかし、いざそのプログラムをどのようにすれば、うまく検索が行えるのかについて検討してみたのだが、あまり良い回答を導き出すことができなかった。現状では、スペースが含まれていれば、無条件に単語を切り離している。とすれば、その作業を行っている箇所に応用を利かせば良いのではないかと考える。しかし、スペースで分けないようにすればいいのだが、それを行う関数を見つけることができなかった。また、正規表現も必要なのかもしれないが、そのあたりに関しても正確な回答を出すことができず、今回達成できなかった点として見送る形にした。

しかし、不可能な機能ではないため、突き詰めていけば必ず実現すると思われる。

(2) 総括

旧サイトを改善することに関して、テーマとして扱った「unicode によるチベット文字での検索・表示」は果たせたと思われる。これは改善するに当たって一番大切なことであったし、利用者の検索のしやすさ、見やすさを飛躍的に進歩させたと考える。また、データベース化に関しても今後のデータの追加や、修正のしやすさに繋がり、非常に有効なシステムになったのではないかと考える。

しかし、その他インタフェースに関して、前項で述べたとおりいくつかの問題点を残すことになってしまった。データベースの不備に関しても同様である。そして、その多くは自身の知識のなさによるものであった。指摘された点においても、有効な機能であると確信しながら、それをどのようにすれば実現できるのかという点において、非常に頭を悩ませる機会が多かった。そのような点を改善することによって、利用者に対しても、管理者に対しても、より便利で、使いやすいシステムになることは間違いないであろう。

また、私自身の問題ではなくコンピュータやブラウザの問題であるものも多く存在した。特にチベット文字の入力や表示などは、すべて OS の機能に依存している。そのため、特定の環境化でなければ正確に確認することができず、また制作中に実際に OS の都合により、チベット文字で入力できず非常に苦慮した。今後、チベット文字が検索・表示できる OS 環境がさらに整い、すべての環境でチベット文字で表示できるようにと期待してやまない。そして、このシステムによって、今後さらに西蔵文献の研究に繋がる事を信じる。

注

（1）『Peking Tripitaka Online Search』

http://web.otani.ac.jp/cri/twrp/tibdate/Peking_online_search.html

文献表

Peking Tripitaka Online Search

http://web.otani.ac.jp/cri/twrp/tibdate/Peking_online_search.html

PHP マニュアル

http://search.net-newbie.com/php/

SQL-TECHSCORE-

http://www.techscore.com/tech/sql/

西蔵大蔵経研究会

1985　『西蔵大蔵経　総目録・索引』臨川書店

大谷大学チベット研究

http://web.otani.ac.jp/cri/twrp/index.html

紙谷歌寿彦

2003　『初めての人のためのかんたん PHP+MySQL 入門』

秀和システム