チベット文献 KWIC 検索システムの構築について

谷本　宏美

目　　次

1　序論

(1) テーマ

私は福田ゼミの「人の役に立つ Web アプリケーションを作成する」というテーマに基づいて、「チベット文献 KWIC 検索システムの構築について」という題目で論文を執筆することに決めた。また、チベット学研究者を対象とした、チベット文献からチベット語の用例を調べることができるオンライン KWIC 検索システムの構築を卒業制作とする。

(2) KWIC サイトの現状

制作物はチベット語の用例を調べることができるシステムを構築する。それは、チベット文字で書かれているテキストファイルから語句を検索して KWIC（クウィック）形式で出力するものである。KWIC とは、KeyWord In Context の略号で、文章データの索引を作成する方法である。この方法は、文章中から指定した語句に加えて、その前後の文脈も同時に表示した索引を作ることで、検索効率を高めることができるものである。

そこで、現在インターネット上で利用できる KWIC 検索サイトを調べてみた。検索エンジンでヒットしたサイトを見ていくと、多くが英語の KWIC 検索サイトであった。そしてチベット語のサイトは 1 つ見つけることができた。実際に制作物を作る際に参考にした 3 つのサイトの特徴を以下に挙げる。

１つ目のサイトは、Web KWIC [(1)]である。このサイトには著作権の切れた文学作品の日本語と英語のテキストが複数登録されており、その中のテキストを 1 件選択して検索語句を入力し、検索を行う。英語のテ

キストであれば正規表現を使うことができる。正規表現とは、メタ文字と呼ばれる特殊な記号を文字列と組み合わせて使い、特定のパターンを指定する方法であり、固定された文字列だけでなく、曖昧な文字列を表現することができる。

そして選択したテキストから検索語句と一致した語句（以下、「一致語句」と呼ぶ）が見つかった場合は、その結果を KWIC 形式で表示する。表示する前後の長さはそれぞれ 40 字であり、1 ページに 10 段落ずつ表示されるようになっている。また、一致語句には色付きで表示され、一致語句をクリックすると該当するパラグラフの内容が詳細として下のフレームに表示される。

このサイトにはメニューがなく、トップと検索結果画面の 2 ページで構成されている。トップは検索入力フォームとサイトの説明、正規表現リファレンスのみを置き、検索結果画面は段落ごとにページを分けて表示するなどシンプルで見やすい構成になっていることから、KWIC システムに特化したサイトだといえる。

2 つ目のサイトは、古代チベット語文献オンライン（Old Tibetan Documents Online）[(2)]である。このサイトに登録されているテキストはチベット文献であるが、全てローマ字に転写されているため、検索語句の入力はローマ字で行う。テキストは複数選択することができ、入力する際は、Web KWIC と同様に正規表現を使うことができる。それに加えて、検索方法として部分一致、完全一致、前方一致、後方一致のいずれかを指定できる。そして検索結果に表示される前後の長さはそれぞれ 50 字であり、ヒットした数に関わらず全て 1 ページに表示される。Web KWIC と同様、一致語句は色付きで表示される。一致語句に該当するページ番号をクリックすると、詳細としてページの内容が表示さ

れる。

3 つ目のサイトは、星研究室 (3)というチベット語について学ぶことができるサイトである。チベット語の動詞辞典があり、動詞辞典で調べた動詞と登録されているテキスト『チベット・ラサの年中行事』と『ミラレパ伝』から一致語句が存在した場合、KWIC 形式で見ることができる。一致語句は他サイトと同様、色付きで表示される。また、検索結果に表示する前後の長さは利用者が希望する長さをプルダウンボックスから選択することができる。詳細のページは用意されていない。

以上のサイトの特徴から、共通点があることがわかる。例えば検索に正規表現が使用できる点、一致語句に色が付いている点、詳細を見ることができる点などである。このような多くの KWIC サイトで使われている機能は、利用者の要望に応じた検索ができ、すぐに知りたい情報を見つけることができるという利点があると考えられる。

一方で欠点は、検索に使用できる文献の制限、そして利用者の制限が存在することである。たとえ、サイトに複数の文献が登録されていた場合でも、利用者が望む文献が必ず登録されているとは限らず、また調べたい語句がそれらの文献に必ず存在するとも限らない。サイトに登録されている限られた文献から、利用者は検索を行うのである。このように文献の制限は、利用者の制限にも繋がる。

また、ここでは説明していないサイトであるが、使用するにあたってユーザ登録を必須とするサイトも存在する。ユーザ登録により、利用者は自分専用のページを持つことができるため、KWIC 検索に限らず幅広く自由に使うことができる。しかし、ユーザ登録やログインには手間が掛かり、誰もが気軽に使用できないため、利用者が制限されてしまうことが考えられる。

最後に、チベット文字の KWIC サイトに関しては、まずチベット文字専用の KWIC 検索ができるサイトを見つけにくいことや、利用できるとしても星研究室の動詞のみを検索できるサイトであり、誰もが気軽に利用できる完全な KWIC サイトは存在しないというのが現状である。

(3) 目標

上記の KWIC システムの現状と利点、欠点を踏まえ、制作物は単なるチベット文献の KWIC システムではなく、チベット学研究者の要望に対応した KWIC システムを制作し、実用化を目指す。そのために、以下の項目の実現を図る。

1. KWIC 形式で表示する
2. 一致語句を赤字で表示する
3. 一致語句のパラグラフ番号を左側に表示する
4. 一致語句のパラグラフ番号をクリックした際、詳細を表示する
5. 正規表現を用いた検索を可能にする
6. アップロード機能を付ける
7. 検索履歴機能を付ける
8. グループでファイルの共有や交換を可能にする

今回チベット学研究者の要望に対応した KWIC システムが完成し実用化されたならば、これまでチベット語の用例を知ることが困難であったチベット学研究者にとって、今後のチベット学の研究、発展に大きく役立つことができると考える。

（4）利用環境

制作物はチベット文字を扱うため、コンピュータ環境においてチベット文字の入力、表示ができることが条件である。チベット文字は 0F00-0FFF ユニコードの制定はされているが、表示環境が十分ではないのが現状である。現在、Windows Vista と Mac OS X Leopard は標準でチベット文字がサポートされている。その他の OS はチベット文字に対応するバージョンやフォントをインストールする必要がある。しかし、インストールをしていても、表示が乱れる可能性が考えられる。

ブラウザについては、Safari では問題なく表示されるが、FireFox では FireFox3、Internet Explorer では Internet Explorer7 が表示可能である。

制作物はチベット文字を表示するため、ユニコードは utf-8 とし、フォントは Kailasa と Microsoft Himalaya、サイズはチベット文字が読みやすいように 20px を指定した。

2　企画

（1）Python を使用する理由

利用環境で述べたが、制作物はユニコードを使用する。制作を開始した際、プログラミング言語は PHP を使用していた。しかし正規表現に問題があり、ユニコードが読み込まれないことから、Python に変更したところ正常に機能した。

PHP はバージョンが変わると関数の戻り値のフォーマットが変わってしまう複雑な言語である。そのため、将来プログラムが対応できなくなったり、何度も中身を書き直さなければならないという欠点がある。

一方で、Python はユニコードによる文字列操作をサポートしており、またシンプルであるため使いやすい言語である。以上の理由から Python を採用した。

(2) 意見の具現化

目標で挙げた制作物の構成や機能について、それぞれの必要性を考えつつ制作を行う。

(i) KWIC 形式で表示する

これが今回の制作で最重要となる部分である。KWIC 形式で表示することは、一致語句とその前後の文章を取り出さなければならない。

まず、CGI[(4)]を使ってトップページ（資料編、図 1 参照）の検索入力フォームで入力した kensaku（検索語句）とチェックボックスから選択した filename（検索対象文献名）のデータを送受信する。そして正規表現の機能を提供する re モジュールを使い、kensaku を正規表現にコンパイル[(5)]する。そして filename を開き、繰り返しの for 文を使って一行ずつ読んでいき、繰り返し正規表現の検索を行う re.finditer() 関数を使い、一致語句を見つける。変数 extend_length で前後それぞれの取り出す長さを指定し、一行ごとに len 関数で一行の長さを数え、表示する前後の長さ分を取り出して print で表示するようにする。

検索結果を表示する際、一致語句の前後の始めと終わりが文字化けとなって表示される問題が生じた。これは、検索入力フォームから送られた kensaku は utf-8 で、kensaku を正規表現にコンパイルしたユニコードは utf-16 であることが原因である。それを同じものにするためには、kensaku の utf-8 をユニコードに変換し、print で出力する際に、encode

で utf-8 に再変換する必要がある。検索語句と一致語句を見つけるために使う正規表現はユニコードで動いていたため、line をユニコードにしたことで、extend_length がきちんと機能するようになり、問題は解決した。

（ii）一致語句を赤字で表示する

一致語句に色を付けることは、どのサイトでも使用されており、どこが一致語句であるか一目でわかるため、制作物でも使用することにする。文字に色を付けるには、HTML 文書の装飾部分を一括管理できる CSS を使用する。一致語句にはクラス指定をして赤色で表示する。（資料編、図 2 参照）

（iii）一致語句のパラグラフ番号を左側に表示する

一致語句が文献のどこに存在するのかすぐにわかるように、結果にパラグラフ番号を表記する必要がある。パラグラフ番号とは、ファイルの行数のことである。パラグラフ番号を調べるためには、ファイルを読み込む際に行数を数える必要がある。ファイルを読み込み始めた時を line_count = 1 とし、line を読み込むごとに line_count に 1 を足していき、一致語句が見つかった場合は、その結果の左側に line の行数を print で表示するようにする。（資料編、図 2 参照）

（iv）一致語句のパラグラフ番号をクリックした際、詳細を表示する

詳細を表示することは、前後の文脈をより深く理解することができると考える。制作物では、一致語句に該当するパラグラフ番号をクリックすると、詳細として該当するパラグラフ番号とその前後の内容が別ウィンドウで表示されるようにする。詳細へのリンクから filename と sw

（一致語句）と line_count（パラグラフ番号）のデータを送る。

そして 3 つのパラグラフを表示するために、ファイルを 1 行ずつ読み込み、繰り返しの for 文を使って該当するパラグラフ番号の 1 行前から 1 行後までの内容を print で表示する。そして検索結果と同様、詳細でも一致語句がある箇所には全て赤字を付けるようにする。（資料編、図 3 参照）

（v）正規表現を用いた検索を可能にする

Web KWIC の特徴の箇所で説明したが、正規表現の使用は、指定された文字列だけではなくパターンを指定することで詳細な検索が可能となる。正規表現は re モジュールを使い正規表現をオブジェクト型にコンパイルしてマッチさせる。

ここで、チベット文字の検索に必要な正規表現はどのようなものがあるかを考えてみる。メタ文字などは、Web KWIC に記載されている正規表現リファレンスを参考にする。Web KWIC では、メタ文字 [6]や文字クラス [7]、量指定子 [8]と多くのパターンが使用できる。そして最後にはパターンの例が載っている。Web KWIC の場合、検索に英語のテキストを使用するため、メタ文字のみでなく、ローマ字や数字も使うことができる。しかし、チベット文字は、検索にローマ字を使う必要はない。そのため、ローマ字のように不要なものは正規表現には使用しないこととする。

その結果、必要な正規表現として以下を選んだ。

．(任意の 1 文字)、*(0 回以上の繰り返し)、+(1 回以上の繰り返し)、?(0 または 1 回の繰り返し)、＾ (行の先頭)、$(行の末尾)、｜ (パターン論理和)、()(グループ化)、[ ](文字クラス)

そのあと福田教授に必要な正規表現について確認して頂き、以下を追加した。

{}(量指定子)、,(量指定子の区切り)、\d(量指定子に使う数字)

(vi) アップロード機能を付ける

利用者が必要とするテキストはそれぞれ違うので、既に登録されているものだけでは利用が制限されてしまうことが考えられる。そのため、利用者が自由にファイルをアップロードできればよいだろう。しかし、単純にアップロード機能を付けるということには問題がある。例えば、これが公開サイト（共有サーバ）の場合、目的から外れた使用をする利用者がいれば、その他の利用者に良くない影響を及ぼしたり、アップロードしたファイルを置くフォルダによってはファイルが重複したりするおそれがある。従って安全で快適に使用できるためには、何かしらの制限を設けることが必要だろう。

安全に利用してもらうためには、例えばログイン画面を付けることが良いかもしれない。そうすれば、自分で自由にファイルを操作することができ、グループを作ることで、グループ内の人とファイルの交換や共有も可能となる。しかし、ログイン画面を付けてユーザ管理を行うには、MYSQL を使う必要がある。また、ログインを行う手間や管理が必要となるため、誰でも気軽に使うことができない。

そこで今回、公開サイトでシステムを利用する場合はアップロード機能は付けず、プログラムをローカル（利用者自身のパソコン）にインストールしてローカルで使う場合はアップロード機能を付けることに決めた。(資料編、図 4 参照）プログラムをローカルにインストールする場合は、settings.py ファイルの UPLOAD を True（アップロード可能）

または False（アップロード不可）を設定する必要がある。（資料編、図 5 参照）

また、ローカルで使う場合は、サーバの軽減化ができるという利点がある。

（vii）検索履歴機能を付ける

検索履歴（以下、「履歴」と呼ぶ）の機能は、何度でも簡単に見返すことができるため、利用者の検索効率を上げることに役立つと考えられる。履歴は検索語句を表示するだけでなく、そこにリンクを付けてクリックすると結果が表示されるように制作する。（資料編、図 6 参照）

まずは履歴が表示できるように、3 つの方法を試みた。

1 つ目はクッキー(9)を使う方法である。検索入力フォームから検索語句のデータが送られる度にデータをクッキーに記録し、その記録を表示することを考えた。しかし、クッキーに記録した検索語句をどこに残し、どれだけの数を記録できるかがわからなかったため、断念した。

2 つ目は tempfile を使う方法である。tempfile は一時的なファイルシステムリソースを作成するモジュールである。利用者がブラウザを開いてサイトに訪問した時に tempfile を作成し、そこに検索語句を追記していくように試みた。しかし、tempfile の保存先が明確でないこと、そして tempfile 内の読み込みや書き込みをする方法が確実でなかったため、実現が困難となった。

3 つ目は履歴用のテキストファイルを使う方法である。これは、検索入力フォームから送られたデータの検索を開始する際、送られた検索語句と文献のデータを rireki.txt（履歴記録ファイル）に追記していく。そして更新を行った際に、リンクを付けた検索語句を表示させ、リンク

をクリックすると検索結果が表示されるようにする。この方法が、3 の中で確実で上手く機能したため、これを採用した。

しかし、文献が複数選択されている履歴をクリックした場合、その履歴の結果が表示されないという問題が生じた。これは、rireki.txt には、選択された文献が 1 件であれば [' 文献名']、複数であれば [' 文献名 1', ' 文献名 2'] というように記録されているが、履歴から結果を表示するリンクは、文献数に関わらず [ ] 内で 1 件の文献であると認識されていたことが原因である。そのため、検索を行う際に文献が存在しないと判断され、結果が表示されなかった。そこで、文字列を特定の文字列で分割してできる配列を作る split() を使い、[ ] 内をカンマごとに区切ることで 1 件ずつ文献ごとに分けることができ、解決した。

また履歴は、フレームを用いて常に履歴を表示しているため、どのページを表示していても使うことができる。しかし、フレームは単純にページからリンクを辿って別のページへ移動しただけでは、履歴は自動的に更新されないため、手動でブラウザの更新ボタンをクリックしなければならない。これは利用者に手間を掛ける上に、気付かない可能性もある。その問題を解消するため、履歴のフレームに JavaScript で作った更新ボタンを置いた。しかし、この場合も同じく手動であることに変わりはないため、60 秒間隔で自動更新を行うようにすることで解決した。

そしてクッキーによる訪問回数を利用し、訪問回数が 1 回目の時、つまり新しくブラウザを開いた時、前回の履歴を消去し新しく履歴を作ることにした。

履歴の表示、そして履歴から結果を表示することは上手くできたが、共有サーバの利用において問題が残った。もし共有サーバで同時に複

数の人が利用していれば、訪問回数も履歴も他者との共有になる。これは、プライバシーが保護されないという問題と、誰か 1 人がブラウザを開けば履歴ファイルが作られ、ブラウザを閉じれば履歴ファイルが消される可能性がある。ローカルの場合は、その利用者のみが自由に使うことができるので問題はない。

そして共有サーバにおける履歴の必要性について考えた結果、履歴を共有することは利用者の要望に反することなど、利用者にとって特に利点は考えられないことから、アップロード機能と同様、プログラムをローカルにインストールした場合のみ使用できることに決めた。従って、settings.py ファイルの RIREKI を True（履歴機能設置する）または False（履歴機能設置しない）を設定する必要がある。（資料編、図 5 参照）

（viii）グループでファイルの共有や交換を可能にする

グループ内で利用者同士が持っているファイルの共有や交換を可能にすることは、文献の制限がなくなることに繋がる。しかしこの場合、ファイルの交換が目的となってしまう。またグループ管理をするために MYSQL の使用やその他の安全面を考える必要がある。今回の制作物は利用者自身が気軽で自由に検索できることを目的としている。これらの目的が異なることから、制作には至らなかった。

3　制作過程

(1) 工夫した点

(i) ページ番号を省いて検索を行う

少なくとも、大谷チベット電子テキストに登録されているテキストの文章には、[A:31.1.4] や [1b] のようにページ番号が含まれている。通常は検索を行う際、そのページ番号が含まれたままでは、ページ番号の前後の語句が一致語句であっても、ヒットするはずがない。

そこで、検索結果をより正確な検索数にするために、正規表現と置換を用いて文章に含まれているページ番号を省いて検索を行うことにした。そして、ページ番号に使われる文字列を推測し、それをコンパイルしておき、ファイルを一行ごとに読み込む際に、ページ番号であると認識された場合、置換を行う sub() メソッドを用いて空文字に置き換える。そのようにすることで、ページ番号を挟んで前後の語句が一致語句であった場合、一致したものとして結果に上手く表示されるようになる。そして検索結果には、ページ番号を省いて表示させる。

上手くページ番号が省かれて検索、表示がされているか、複数の文献から検索を行った結果、どの文献でも検索結果に正確に表示されていることがわかった。

(ii) 前後の長さの選択

星研究室の KWIC サイトでは、一致語句の前後の長さの表示を「短い」「標準」「長い」のいずれかを選択することができる。これは、利用者の好みによって選択できるので、快適にシステムを利用することができる機能であると考える。

制作を始めた際、チベット文字の 1 文字は組み合わせの数からなる文字数であるということを踏まえ、前後の長さはそれぞれ 100 文字ずつ表示するようにしていた。しかし、それでは表示に無駄や不快さを感じる利用者がいるかもわからない。しかし、実際にどのくらいの長さが適切であるかわからなかったため、星研究室の KWIC 検索を参考にして、制作物もプルダウンボックスを使い、「短い（40 字)」「標準（60 字)」「長い（80 字)」のいずれかを選択できるようにした。

(iii) エラー処理

利用者が快適にシステムを利用してもらえるように、プログラムを正常に動かせない場合は、エラー処理やエラー表示をしなければならない。エラー処理は 2 通りある。

１つ目は、例外処理である。例外が起こる可能性がある処理の範囲をマークした上で例外発生時の処理を別途記述することにより、プログラムを終了させることなく、例外発生時にそれを捉えて適切な処理をさせることが可能である。Python は例外処理の try,except というモジュールがある。try ステートメントを利用した場合、そのブロック中でエラーが発生すると except ブロック中の例外処理が実行される。例外処理後は try ブロックの後のコードから実行される。

制作物は、検索入力フォームに何も入力されていない場合、存在しないファイルを選択した場合、文献のユニコードの問題で内容が読み込めない場合、などエラーが起こる可能性がある箇所に例外処理を行い、それらのエラーが発生した場合、「チベット文字を入力してください」「選択したファイルが存在しません」などのエラーを表示するようにした。そしてその後に記述している通常のプログラムの処理が始まらないよ

うに、プログラムを終了させる sys.exit() を記述した。(資料編、図 7 参照)

2 つ目は、検索語句の入力に対するエラー表示である。このシステムはチベット文字専用であるため、入力の条件としてチベット文字が含まれていることが必須である。またチベット文字に加え、メタ文字の使用も可能である。もし、条件に合っていない場合には検索ができないことを伝える必要がある。そこで、送られた検索語句のデータがチベット文字であるかどうかを判断するために、re_tibetan = re.compile(u"[\u0F00-\u0FFF\d\,*\?\ | \ ^ $\.\+\(\)\[\]\{\}]\+$")(チベット文字、またはメタ文字)とし、正規表現が先頭で一致するかを調べる match() 関数を使い、if not re_tibetan.match(u_kensaku):という検索語句が re_tibetan と一致していないと判断した場合は、「チベット文字で入力してください」「正規表現に間違いがあります」など入力した文字に合わせたエラーを表示するようにした。

しかし、できるだけ分かりやすくするため、前述の条件を簡略化することにした。re_tibetan = re.compile(u"[\u0F40-\u0F68]") (ツェクやシェーを含まないチベット文字のみ) とし、文字列の全体からマッチする部分を探す search() 関数を使い、if not re_tibetan.search(u_kensaku): という検索語句が re_tibetan を含んでいないと判断した場合に、エラー表示をするように変更した。(資料編、図 8 参照)

これらのエラー処理、エラー表示により、利用者がすぐに状況を理解できるように工夫した。

(2) 問題点

制作過程においていくつか問題が生じた。問題が解決していないものについては、今後の課題となる。

(i) 有足字の表示

チベット文字は、複数の文字が合成して形成する複雑な文字である。しかしここでは、1 音節が 1 文字としてカウントされるのではなく、1 音節を形成するために組み合わせている数が文字数カウントされる。

そして、1 音節の下部に付く有足字における表示に問題が生じた。それは、検索語句の最後の文字に対して有足字が付く場合、検索結果には有足字が一致語句の横に表示されてしまうのである。詳細でも同様に表示される。

この問題は、検索語句の文字を表示する順序がチベット文字の 1 音節を書く順序と関係があるからだと考える。つまり、検索結果で検索語句まで表示したあと、それ以降の順序である有足字は別の文字だとみなされ、隣に表示されてしまうのだ。この問題は、1 音節であることを無視し、正確なチベット文字を表示していないことから、利用者にとって読みにくさを感じさせるだろう。

そこで、この問題を解決するために有足字、または有頭字がくる可能性がある文字をコンパイルしておき、文章中から一致語句を見つけた場合、一致語句の次の文字が有足字、または有頭字であるかを調べる。もしそうであった場合は一致語句ではないと判断し、表示しないようにすることで解決した。(資料編、図 9,10 参照)

(ii) ツェクを含む正規表現

メタ文字にツェクが関わっている場合、検索は正確に行われるが、詳細を表示した際に問題が生じる。

例えば、[ ]（文字クラス）内にツェクを使用していた場合、詳細の全体がエラーで表示されたり、ツェクに対して繰り返しを意味するメタ文字を使用していた場合、一致語句であるにも関わらず赤字で表示されない。

その原因として、一致語句がページ番号を挟んでいる場合、詳細にはページ番号を含めて赤字で表示するようにしていることが挙げられる。一致語句がページ番号を挟んでいる場合の置換方法は、以下である。

u_kensaku = u_kensaku.replace(u"\u0F0B", u"\u0F0B?(?\[[\w \.\,\:\-\;]+\] | \([\w \.\,\:\-\;]+\) | \[\w \.\,\:\-\;]+\?)* ?")

これは、一致語句のツェクの後ろに正規表現で指定しているページ番号がくる場合、一致語句としてグループ化するようにしているが、そこでツェクを使用しているため、上記の問題が生じてしまうのである。従って、現時点では以下のどちらかを選択することで問題を減らすことに決めた。

1. ツェクに対してメタ文字を使用した場合、一致語句に赤字で表示されるが、ページ番号を挟んでいた場合は表示されない。
2. 一致語句がページ番号を挟んでいた場合、ページ番号を含めて赤字で表示されるが、ツェクに対してメタ文字を使用した場合は表示されない。

まずは詳細がエラーせずに表示することを優先するため、1 の方法に決めた。そのため、上記の置換の記述をやめ、ページ番号を挟んでいな

い一致語句の場合のみ、赤字で表示するようにした。しかし、どのような場合においても一致語句は赤字で表示することが適切であると考える。そのため、今後は改善が必要だろう。

(iii) 他文献でも対応できるか

現在制作物に登録されている文献は、大谷大学真宗綜合研究所の西蔵文献研究班の大谷チベット電子テキスト[10]から頂いたもので、主に歴史書物が多い。試行ではそれらの文献を使用し、その文献の形式に合わせて検索できるように制作している。そのため、アップロードしたファイルの検索を行うと、文献の種類、形式によってはどのように表示されるかはわからない。

そこで、福田教授からそれまでの形式とは異なる仏教に関する文献を頂き、実際にアップロードと検索を行った。その結果、上手くアップロード、検索、そして表示ができたことを確認した。

(iv) アップロードファイルの可能性

複数の拡張子のファイルをアップロードした際、テキストファイル、HTML ファイル、word 文書ファイルなどは上手く検索、表示できることを確認した。しかし、文書の文字そのものがひとつの表現となるリッチテキストファイルはチベット文字が読み込めず、検索、表示をすることができなかった。

また、ファイル名については、ローマ字と使用可能な記号であれば検索を行うことが可能である。しかし、日本語やチベット文字を使用したファイルは、アップロード後にファイル名が読み込めず、中身を表示することができなかった。

今回は複数のファイルから上手く作動するかを試してみたが、その数

は限られている。このことを踏まえ、もしアップロードしたファイルが検索に対応していなかった場合は、ファイルの種類を変えてもらうか、もしくはそれらのファイルに対応するようにプログラムを作るか、決めなければならない。より多くの文献を使用してもらえるために、今後も文献を手に入れて調べてみる必要がある。

(v) アップロードに同名ファイルが存在する場合

アップロードしたファイルが増えていくと、ファイルの管理が困難となり、すでにアップロードしたファイルと同名のファイルをアップロードしてしまうことが考えられる。そしてファイルによっては、利用者の意に反して上書きされてしまうことが不都合な場合もあるだろう。そこで、アップロードするファイルと同名のファイルが存在するかを調べ、同名ファイルが存在する場合は上書きをせず、中止メッセージを表示させるようにした。

しかし、現時点のアップロードファイルの問題点として、アップロードの失敗が挙げられる。これは、アップロードに時間がかかりブラウザの中止ボタンや更新ボタンをクリックした場合、検索画面にファイル名が登録されるが中身は空で登録されてしまうのである。そして、もう一度同じファイルをアップロードする場合や、不要なファイルを削除したい場合は、ローカルディスクの文献が入っているフォルダからファイルを選択して削除を行わなければならないという欠点がある。

この欠点をなくすため、同名ファイルが存在した場合はそのファイルを「上書きする」「別名にする」「やめる」のいずれかを選択できるようにする必要があると考えた。

そしてこれらの機能を付けるために、フォーム要素を使い、各選択

肢を作った。「やめる」に関しては、ページの更新ボタンにすることで対応できた。また、「上書きする」「別名にする」に関しては、CGI と Javascript を使用する方法を試みた。「上書きする」を選択した場合は、CGI でファイル名を送り、通常のアップロードと同様にアップロードを開始する。また、「別名にする」を選択した場合は、os.rename() を使い、名前を変更してからアップロードを開始する。しかし、これらの方法はフォームから元のファイル名とファイルの中身を上手く送ることができず、今回実現するに至らなかった。

結局、アップロードに失敗する可能性があることと、アップロードしたファイルの元ファイルは自身のローカルに残っている可能性が高いことから、現時点では同名ファイルが存在した場合、アップロードを中止することはやめ、上書きすることに決めた。しかし今後の課題として、上記の機能を付ける必要があるだろう。

(vi) アップロードファイルによる登録数の制限

現在、制作物には 10 件の文献が登録されている。そして今回、ファイルのアップロードが可能となったことで、文献の制限を大きく変えることができた。

しかし、文献の制限がなくなることは検索の制限をなくすことにも繋がるが問題も生じる。例えば利用者が多くのチベット文献ファイルを持っており、それら全てをアップロードした場合、利用しづらい環境になってしまうという問題が挙げられる。具体的にいえば、ファイルの管理が困難となったり、検索入力フォームや文献一覧画面で 1 行に 1 件ずつ表示している文献名が縦に並ぶため、スクロールが必要になるなど、検索の効率を下げることに繋がってしまう。従って、今後の課題とし

て、ファイルの制限を設けることと表示方法を改善することが必要だと考える。

4　結論

(1) ベータテストの結果

実際に、チベットの研究をされている大谷大学真宗綜合研究所・西蔵文献研究班所属の三宅講師と同所属の大学院生に数回ベータテストをして頂き、多くの意見を頂いた。それらの意見に対する解決策は以下の通りである。

・「検索結果の表示に、ページ番号を消した代わりに、ツェクが表示されている。」

この意見に対してコードを確認したところ、検索でページ番号を空白に置き換える際の正規表現を、ツェクと記述していたため訂正をした。

・「検索結果画面に移った際、検索入力フォームに検索語句の入力と文献名チェックを入れてほしい」

検索結果画面には、トップと同じ検索入力フォームを置いている。意見も頂いたときは、検索結果画面の検索入力フォームには何も手を加えず空白のままであった。

しかし、それでは検索語句と選択文献がわからなくなる、再び入力しないといけない、という意見を頂いたため、検索入力フォームに検索語句と、選択した文献名を表示することにした。そのために、送られた kensaku のデータを検索入力フォームに value = kensaku と入れることで、送られた検索語句が入力されるようになった。次に選択された文献は、送られた filename のデータが文献を入れているフォルダ内の

ファイル名と一致した文献であれば、チェックボックスにチェックを入れ、そうでない場合はチェックを入れずにファイル名を表示することで解決した。

・「履歴は検索語句だけでなくテキスト名も表示してほしい」

この意見に対して、私自身も履歴に検索語句のみを表示してもどの文献を選択したのかわからないと感じた。しかし、検索語句と並べて文献も表示すると全体の履歴画面が見づらくなってしまう。そこで、フォームのリンクに title を使い、検索語句にマウスを乗せると選択した文献名を表示することで解決した。

・「正規表現の説明が難しい」

この意見に対して、ベータテストを行った時点では、正規表現リファレンスには使用できるメタ文字とその意味のみ書いていた。意味は短く書くことで、読みやすく分かりやすいだろうと思っていたが、逆効果であった。この検索システムを利用する人は、正規表現を使うことに慣れている人もいれば、正規表現とは何なのか、初めて言葉を聞くという人もいるだろう。正規表現とは何なのか知らない人でも、すぐに理解して詳細な検索が行えるように、「正規表現とは」という説明を加えた。そして正規表現リファレンスには、メタ文字とその意味に加え、チベット文字とメタ文字を使った入力とその結果の例を挙げ、難解な部分には赤字で注意書きを入れる工夫をした。(資料編、図 11 参照)

・「詳細には、検索語句ではなく一致語句だけが赤字で表示される」

正規表現を使用して検索を行った場合、結果に表示される一致語句はそれぞれ同一語句であるとは限らない。これはそれらの結果から詳細を表示した際、検索語句ではなく一致語句のみが赤字で表示されることに対する意見である。つまり、詳細に表示される 3 パラグラフ内に、検索

語句が存在しても、リンク元と同じ一致語句のみ赤字で表示されないのである。正規表現を使用して検索を行ったにも関わらず、特定の一致語句のみ赤字で表示されることは問題である。一致語句のみ赤字で表示される原因は、詳細へ飛ぶリンクが sw（一致語句）のデータを送っていたためである。従って、そこを u_kensaku（検索語句）と変更することで、一致語句のみでなく、入力した検索語句に一致する語句全てが赤字で表示されるようになった。

・「フレームの表示が汚い」

履歴を表示するフレームを下にスクロールした時、複数の黒い線が表示されてしまうことがある。これはフレームが非推奨であるためバグが起こることがあるのだ。バグを起こさないためには、フレーム、またはラインを使用しないことが望ましい。しかし、履歴を常に表示させて置きたいこと、そして作成段階で全てのページにメニューを置くという手間が掛からないことから、フレームを使うことを選んだ。

(2) 自己評価

今回の制作において、目標の各項目に挙げた制作は達成できたといえる。現存する KWIC サイトの欠点であった制限をなくすために新しい機能としてファイルのアップロードを可能にしたり、利用者の要望に対応した選択制の機能を備えることなど実現できたことから、現存する KWIC サイトの差別化も図れただろう。

しかし、問題点で述べた課題が残っている。今回達成できた機能に加え、より安全で使いやすい制作物にするために、今後も改善に取り組む必要がある。例えば、アップロードファイルに対して「上書きする」「別名にする」「削除する」の各機能を付けることが必要だと考える。アッ

プロード機能という他のサイトにはない機能を加えたとはいえ、サイト上で簡単にファイルの操作ができないのでは、利用者に負担がかかってしまうだろう。

また、英語や日本語に対して捉え方が異なるチベット語の仕組みについて、もっと理解しておく必要があった。そのことが、チベット文字自体の問題と正規表現を使用した時の問題として現れている。私自身はこれまでにチベット文字を学んでいなかったため、検索結果におかしな点を気づくことができなかった。しかし、実際にベータテストにおいて複数のパターンで検索を行って頂いたことにより、検索結果に有足字が離れて表示されるなど、複数の問題箇所を見つけることができた。

他に、チベット文字の 1 音節を形成するために組み合わせた文字の数が文字数となることから、メタ文字の [ ] を使用した場合、[ ] 内の文字が部分ごとに分かれて検索を始めてしまうことがわかった。この問題は、正規表現の注意点として正規表現リファレンスで「組み合わせの数が文字数になります」と注意書きをしておいた。しかし、これらの問題を訂正しないと、利用者が気軽に使える検索システムとは言い難いだろう。また、実際に説明が不十分なために分かりづらい、見えにくいなどの意見を頂き、改めて利用者に対して使いやすいサイトについて考えることができた。

今回生じた問題は、今後実用化を実現する上で重要な課題であるといえる。

注

(1) http://app.yohasebe.com/kwic/index.php

(2) http://otdo.aa.tufs.ac.jp/

(3) http://star.aa.tufs.ac.jp/

(4) Web サーバが、Web ブラウザからの要求に応じて、プログラムを起動するための仕組み。

(5) 人間がプログラミング言語を用いて作成したソフトウェアの設計図 (ソースコード) を、コンピュータ上で実行可能な形式 (オブジェクトコード) に変換すること。

(6) その文字本来の意味とは異なり、プログラムで特別な意味を持たせた文字のこと。

(7) 正規表現の中で該当する位置に一致可能な文字を表すリストを指定するために使用する。文字クラスを定義するには、リストを角括弧（[および]）で囲む。

(8) 指定した文字が繰り返し出現するパターンを作成したい場合に使用する。繰り返す回数を指定するには {} に回数を指定する。

(9) Web サイトの提供者が、Web ブラウザを通じて訪問者のコンピュータに一時的にデータを書き込んで保存させるしくみ。

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

文献表

Python.jp　http://www.python.jp/Zope

Web KWIC　http://app.yohasebe.com/kwic/index.php

——チベット文献 KWIC 検索システムの構築について——

古代チベット語文献オンライン　http://otdo.aa.tufs.ac.jp/

星研究室　http://star.aa.tufs.ac.jp/modules/bluesbb/viewtopic.php?topic=1

大谷大学チベット研究　http://web.otani.ac.jp/cri/twrp/index.html