社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－DBS－136 (1)
2005－ＦＩ－ 79 (1)
2005／5／19

非ローマンアルファベット系言語の原綴り・翻字相互変換システムの構築

望月 源† 大和 加寿子‡ 前嶋 淳子‡ 林 俊成†
†東京外国語大学 外国語学部
‡東京外国語大学 附属図書館
{motizuki, yamato_kazuko, maejima_junko, lin}@tufs.ac.jp

[概要]
本研究では，非ローマンアルファベット系言語の図書資料を扱う図書館での書誌情報の登録を効率化し，蔵書検索の利便性を向上させるための原綴りと翻字の相互変換システムを作成する．本稿では，翻字規則としてALA-LC 方式の翻字規則を採用し，開発言語としてキリル文字を使用するロシア語とデーヴァナーガリー文字を使用するヒンディー語を選択し，原綴りから翻字への自動変換を行なう手法，翻字から原綴りへの自動変換を行なう手法について述べる．書誌情報を評価データとして用いた実験によりシステムの評価をする．また，翻字規則と言語の違いによる翻字変換の難易度についても述べる．

キーワード 書誌情報， 非ローマンアルファベット， ALA-LC 翻字， 多言語蔵書検索

Development of Mutual Transcription System between Languages that use Non-Roman Alphabets and Their Romanizations

MOCHIZUKI Hajime†, YAMATO Kazuko‡, MAEJIMA Junko‡, LIN Chun Chen†
†Faculty of Foreign Studies, Tokyo University of Foreign Studies
‡Tokyo University of Foreign Studies Library

Abstract
In this paper, we aim to develop a mutual transcription system between languages that use non-roman alphabets and their romanizations. We require the system to have two functions; the function that transcribes original language's strings into their romanized transliterations, and the function that converts roman transliterations to their original language's strings. We adopted Russian which uses Cyrillic and Hindi which uses Devanagari alphabet as original languages and the ALA-LC romanization tables as transliteration rules for our first system. In this paper, we describe a method to romanize non-roman languages automatically according to ALA-LC rules and a method to reconstruct original language's strings from ALA-LC transliterations automatically. We evaluate our system by examinations. We can apply the system to make two application systems; a registering system which is capable to register book information to the online book catalog by ALA-LC romanizations efficiently, and a library retrieval system which is capable to retrieve books from romanization database by using original language's scripts.

keywords book catalog database, Non-Roman Alphabet, ALA-LC Romanization table

1. はじめに

近年，計算機での多言語処理の環境は大幅に向上し，扱える言語の数も増えている．しかし，計算機利用に関して長い歴史のある図書館蔵書検索や書誌情報の登録では，非ローマンアルファベット(以降「非ローマ字母」と記す)系言語の綴り字をそのまま利用することはあまりなく，伝統的にローマンアルファベット(以降「ローマ字母」と記す)での翻字が用いられている．ここでいう翻字とは，「ローマ字母による転写」のことであり，ある言語の音韻を考慮したうえで，文字レベルだけでなく文脈などの情報も加味した発音をローマ字母で書き表すことを言う[1]．この翻字規則には様々な方式が存在し，言語ごとに個別に定義されているが，日本では，多数の言語に対してALA-LC方式の翻字規則が一般的に用いられている．

原綴りを直接用いるのに比べ，翻字化には人的・時間的コストがかかる．そのため，非ローマ字母系言語の書誌情報の登録作業は効率があまりいいとは言えない．また，検索時にも翻字を用いる必要があるが，一般利用者にとって翻字は馴染みがないため，蔵書検索の利便性も悪くなる．さらに，検索結果も翻字で表示されるため，翻字から元の言語で何と書かれているかを理解しなければならない．理想的には，あらゆる言語について，書誌情報を原綴りで作成し，翻字なしで処理が行なえるようになることが望ましい．しかしながら，一部の非ローマ字母系言語では文字コードが充分に整備されていない，フォントが充分でないという問題が依然として存在する．将来的にUnicode[2]などの普及により，広く原綴りの利用が可能になったとしても，諸言語による書誌情報を一元的に扱う場合の整合性を確保する必要やこれまで蓄積されたデータの有効利用のためにも，ローマ字母による翻字は原綴りと併用され，当面主要な方法として利用されるものと思われる．つまり，翻字の必要な図書資料を扱う図書館においては，今後も翻字の処理の効率化は重要な問題であり続けると考えられる．

東京外国語大学では外国語学部において２６言語[2]の専攻語教育を行なっており，附属図書館では非ローマ字母系言語も含めた多様な言語による図書資料を日常的に扱っている．こうした事情もあり，現在我々は，次の２つを実現する非ローマ字母系言語の原綴り・翻字相互の自動変換システムを開発している．

- 原綴りから翻字への自動変換
- 翻字から原綴りへの自動変換

前者の利用により，書誌情報の翻字での登録に際して，作業者が原綴りを最大限利用できるように支援する「書誌情報登録支援システム」と，一般の利用者が原綴りで検索を行なえる「蔵書検索システム」が実現できる．一方後者の利用により，「書誌情報登録支援システム」において，既存の書誌情報内の翻字から原綴りを作り，書誌情報に追加する機能が実現可能になる．また，「蔵書検索システム」の翻字での検索結果を原綴りに変換し，利用者の利便性の向上も計れる．

本稿では，ALA-LC方式の翻字規則を対象とした原綴り・翻字の相互変換システムについて述べる．また，最初に実装する言語として，比較的難易度の低いと考えられるロシア語およびヒンディー語を選びシステムを構築した．構築した変換システムの精度を確かめるため書誌情報を用いた実験を行ない評価する．

以下２節で，翻字規則と言語の違いによる翻字の難易度について述べる，３節で，本稿で扱う原綴りと翻字の相互変換システムについて説明する．次に，４節で，原綴り・翻字変換システムの精度を測るための実験を行なう．

2. 翻字規則と変換の難易度

本節では，翻字規則について簡単に説明し，翻字の言語毎の難易度について検討する．

2.1. 翻字規則

原綴りのローマ字母での翻字は，前述したように，ある言語の音韻を考慮したうえで，文字レベルだけでなく文脈などの情報も加味した発音をローマ字母で表すことを言う．様々な翻字方式の中から，どの方式の翻字規則を用いるかは基本的に各図書館の自由であるが，書誌情報の共有化の観点から，互いに交流のある図書館どうしでは同じ翻字規則を用いることが一般的である．日本では，多くの図書館が国立情報学研究所の目録所在情報サービス(NACSIS-CAT[4])に参加しており，書誌情報の共有を行なっている．

NACSIS-CATでは，非ローマ字母系言語の書誌情報登録は，原綴りのみを登録する言語(日本語，韓国

---

[1]より狭い意味での翻字を捉える場合は，「ある言語の通常の文字体系で表記されたものを，他の文字体系に移し換えることを指す．もとの文字と変換先の文字とを規則的に対応づけることを基本とし，理想的には，翻字された文字からもとの文字への再変換が，機械的な置き換えだけで一義的におこなえるものである」ことを言う[6]が，本稿では広い意味で翻字を捉える．

[2]英語，ドイツ語，フランス語，イタリア語，スペイン語，ポルトガル語，ロシア語，ポーランド語，チェコ語，中国語，朝鮮語，モンゴル語，インドネシア語，マレーシア語，フィリピン語，タイ語，ラオス語，ベトナム語，カンボジア語，ビルマ語，ウルドゥー語，ヒンディー語，アラビア語，ペルシア語，トルコ語，日本語．

語[3]，原綴りと翻字を登録する言語(中国語，アラビア文字使用言語)，翻字のみを登録する言語(その他の言語)に分けられる．翻字を登録する言語の内，中国語ではピンインが用いられるが，残りのすべての言語では ALA-LC 方式の翻字規則が使用されており，ALA-LC 方式が事実上の標準規則といって良い．中国語のピンインは翻字としてそれほど負担がかからないため対象とせず，本研究では ALA-LC 方式を対象とする．

ALA-LC 方式翻字規則(ALA-LC Romanization Tables)は，全米図書館協会(the American Library Association)と米国議会図書館(The United States Library of Congress)が定めた翻字規則の集合である[3]．150 言語以上の非ローマ字母系言語を音標符号付きのローマ字母に置き換えるための 54 種類の原綴り翻字対応表が定義されている．ALA-LC 方式での翻字変換は，基本的には対応表に従って綴り字を音標符号付きのローマ字母に置き換えることで行なう．ただし，発音を重視した翻字であるため，単純な文字レベルの対応表だけでなく，文脈などによる発音の変化に合わせた例外的な規則がどの言語の場合にも存在し，より複雑なものとなっている．この複雑さの度合は，その言語に必要な例外的な規則の種類や数の違いによって異なり，翻字化の難易度に関係する．一般的には同じ綴り字を何通りにも読み分ける言語や，母音を補って読む必要がある言語などの翻字は，より複雑で難しい．

言語によって翻字規則の内容が異なるため，原綴り・翻字自動変換システムの主要な部分は個別の言語ごとに実装する必要がある．次の 2.2 節で，実装の候補となる言語における原綴り・翻字変換の難易度について述べ，最初のシステムとして実装する言語を決定する．

2.2. 変換の難易度と実装言語の決定

2.1 節で述べたように，ALA-LC 方式の翻字規則は，言語によって大きく異なるため，原綴り・翻字変換システムの実装では，主要な部分を各言語ごとに開発する必要がある．そのため，システム全体としての共通インタフェースを与える部分と，実際に個別の言語を処理する部分を分けることが現実的である．そこで本研究では，各言語固有の部分をモジュールとして実装することとし，附属図書館で日常的に扱っている非ローマ字母系の言語，ロシア語，モンゴル語，ヒンディー語，アラビア語，ペルシア語，ウルドゥー語，タイ語，ラオス語，カンボジア語，ビルマ語の中から実現難易度や必要性などを考慮して，モジュールとして実装する言語とその実装順を決定することとした．具体的には次の方針でモジュール化する言語を選択した．

- 各候補言語について，原綴り・翻字間での変換の難易度を調査する．
- 現時点で実装が可能であるものについて難易度の順位付けをする．
- 図書の流通量なども考慮し，実装する言語を選ぶ．

2.2.1. 難易度の調査

2.2 節に示したモジュール実装の候補となる 10 言語について，各言語の特徴や ALA-LC 方式の翻字規則を参照し，次の点に注目して原綴り・翻字変換の難易度を調査した．

- ALA-LC 方式の翻字規則中の綴り字と翻字の数．ALA-LC 方式の翻字規則に記された原綴りと翻字の対応表での原綴りの数と翻字の数．一般に元の言語の文字数と翻字数が同数に近い方が，原綴りと翻字の一対一対応の割合が増え，曖昧性が少なくなるため，自動化処理の難易度が下がると考えられる．また，翻字の数が少ないものは，異なる綴り字が同じ翻字になる割合が高いため，翻字から原綴りへの変換時に曖昧性が大きくなる．原綴りの文字数が少ないものは，原綴りから翻字への変換時に曖昧性が大きくなると考えられる．
- 例外的な扱いの必要な翻字規則の多さや複雑さ．翻字表の中に記された例外規則数の多少と，例外の内容の複雑さ．一般に例外の数が多いほど自動化処理は難しくなる．また，例外の内容が複雑なものであれば，数が少なくとも難しい場合もある．
- ALA-LC 方式の翻字規則の記述ページ数．ALA-LC 方式の翻字規則で，その言語についてどのくらいのページ数を割いているか．一般に，ページ数の多い言語は自動化処理が難しい．
- 元の言語（原綴り）における分ち書きの有無．その言語が，分ち書きをするかしないか．ALA-LC 方式の翻字規則では分ち書きを行なう必要がある．そのため，分ち書きのない言語では，意味の切れ目で分ち書きをした上で，翻字化する必要があり，それだけで自動化処理が難しくなる．
- Unicode の整備状況，計算機での入力可否．Unicode4.0.0[4]において，文字コードやフォントが整備されているか，および，Windows での文字入力が可能かどうか．文字コードが規定されていても文字が不足している場合やフォントが整備

[3]韓国語では，マッキューン・ライシャワー式翻字[1]や韓国文教部で定めたローマ字表記[4]などが用いられるが，NACSIS-CAT ではオプションであるため，翻字が利用されないことも多い．

[4]原稿執筆時の最新バージョンは 4.1.0 であるが，本研究で言語について検討を行なった時点では，4.0.0 が最新であった．

表 1：翻字の難易度に関する調査結果

言語	文字	綴:翻	頁	例外規則	分かち	その他
ロシア	キリル	76:76	2	単純，少	あり	
モンゴル	モンゴル	126:78	2	やや複雑	あり	Unicode 文字不足
ヒンディー	デーヴァナーガリー	88:94	4	やや複雑	あり	
アラビア	アラビア	132:46	10	複雑，多	あり	母音表記なし，大文字・小文字区別なし
ペルシア	アラビア	130:41	7	複雑，多	あり	
ウルドゥー	アラビア	181:65	8	複雑，多	あり	
タイ	タイ	102:86	16	複雑，多	なし	
ラオス	ラオ	79:83	4	やや複雑	なし	Unicode 文字不足，Windows での入力困難
カンボジア	クメール	118:78	3	やや複雑	なし	
ビルマ	ビルマ	91:85	3	やや複雑	あり	

されていない場合などもある．また，現在最も普及している OS である WindowsXP において，特別なソフトウェアを必要とせずにその言語の文字入力が可能かどうかも実質的な利用のしやすさに関係し，実装対象としての判断材料となる．各言語についての調査項目とその結果を表 1 に示す．

表 1 より，現時点で文字コードが完全とは言ない，モンゴル語，ラオス語，カンボジア語，ビルマ語は現時点での実装が難しい．また，翻字には分ち書きが必要であるが，タイ語，ラオス語，カンボジア語は，分かち書きがされない言語である．分ち書きの自動化には形態素解析が必要になるが，それ自体が難しい研究課題となるため，現時点での実装は難しい．

残る 5 言語は，文字としては 3 つのグループ，キリル文字，デーヴァナーガリー文字，アラビア文字に分けることができる．この中で，アラビア文字を用いる言語の場合には，原綴りで母音を用いないため，翻字を行なう際に文脈から母音を補う必要がある．また，アラビア文字を用いるどの言語においても，翻字規則の中に例外に関する複雑な記述が多く，相対的な難易度が高い．総合的に検討すると，キリル文字，デーヴァナーガリー文字，アラビア文字を使う言語の順に翻字の相対的な難易度は低いといえる．結果として，ロシア語(キリル文字)とヒンディー語(デーヴァナーガリ文字)が比較的実装が容易であると考えられる．最終的に，図書資料としての流通量も考慮して，今回はロシア語と，ヒンディー語によるシステムを構築することにし，アラビア文字を用いる言語については，次の開発言語とすることにした．

3. 原綴り・翻字自動変換システム

本節では，我々の構築した「原綴り・翻字相互変換システム」について説明する．

3.1. 自動変換システムの概要

我々が開発している非ローマ字母系言語の原綴り・翻字相互の自動変換システムのイメージを図 1 に示す．

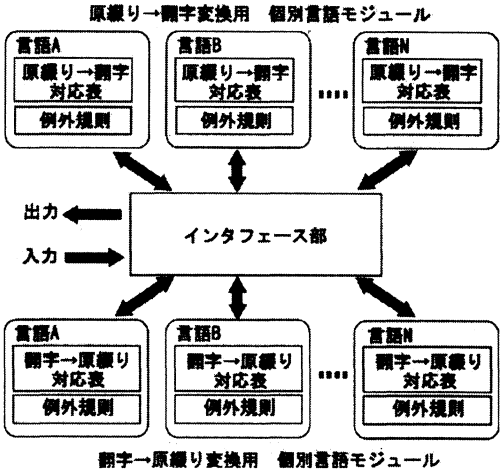

図 1：自動変換システムイメージ図

本システムは，システム全体としての共通インタフェースを与えるインタフェース部と，各実装言語に依存して実装されるモジュールから構成される．インタフェース部では，入力文字列をその言語に対応した変換モジュールに渡し，その結果を受け取りユーザに返す．一方各言語モジュールは，「原綴りから翻字への変換用モジュール」と「翻字から原綴りへの変換用モジュール」に大きく分かれ，それぞれ各言語用に機能する．「原綴りから翻字への変換用モジュール」は，各言語ごとに翻字化の規則(「原綴り→翻字対応表」)と例外規則からなる．「翻字から原綴りへの変換用モジュール」は，各言語ごとに原綴り化の規則(「翻

字→原綴り対応表」)と例外規則からなる．各言語用のモジュールを変換方法別に作成し，追加することによって対応する言語を増やせるようになっている．

本システムは以下の2つの自動変換機能を提供する．

- 原綴りから翻字への自動変換．ある言語の原綴りを，それに対応するALA-LC方式の翻字に変換する機能．
- 翻字から原綴りへの自動変換．ある言語のALA-LC方式の翻字を，それに対応する原綴りに変換する機能．

次副節では，ロシア語とヒンディー語での各変換機能の詳細について説明する．

3.2. 原綴りから翻字への自動変換

原綴りから翻字への変換は，各言語ごとにALA-LC方式の翻字規則に則って行なう．翻字規則の中で，原綴り側の各文字を翻字に対応させるだけで処理できる部分については，「原綴り・翻字対応表」としてまとめる．一方，前後の文字の種類や音韻などを考慮して場合分けしながら翻字を決定する必要のある部分については，個別に「例外規則」として作成する．

この対応表と例外規則による原綴りから翻字への自動変換の基本的な処理手続きは次のようになる．まず，原綴りの先頭から1文字を取り出し，その文字を含む例外規則が存在するかどうかを調べる．もし，例外規則が存在する場合は，その文字の位置や，前後の文字も調べ（言語や例外の種類によりどれだけの範囲を見る必要があるかは異なる），例外に該当するかどうかを判定する．例外に該当する場合は，その例外規則によって翻字を行なう．該当しない場合，および，例外規則に関連がない場合は，対応表に基づいて文字レベルでの翻字を行なう．該当する翻字のない文字はその言語以外の文字か記号であると判断し，翻字しない．

なお，ALA-LC方式の翻字規則はその言語の基礎的な知識がある人間向けに書かれているため，記述内容を解釈しながら，計算機上で実現できる形で規則化する必要がある．本研究では各言語の書誌情報登録作業者の助言を得ながら，具体的な規則を作成している．

3.2.1. ロシア語

ロシア語のALA-LC方式の翻字規則には，大文字，小文字それぞれ38文字，計76文字[5]に対応する翻字が定義されている．本研究では，これらの文字について，原綴り・翻字対応表を作成した．

また，「ъが単語の最後に出現する場合は，翻字しない」という例外が1つある．本研究ではこの例外を扱うため次の例外規則を作成した．

- ъが出現する場合にその文字の単語内の位置を調べ，最後の場合は翻字化せず，最後でない場合は翻字化する．

なお，規則の動作確認のため，約200の原綴り・翻字データを作成し，繰り返し規則の修正を行なった．

3.2.2. ヒンディー語

ヒンディー語のALA-LC方式の翻字規則では，音節の先頭にくる母音・二重母音19字，子音に続く母音・二重母音19字，子音46字[6]およびサンスクリット語からの外来語で用いられる文字である，アヴァグラハ(Avagraha)「ऽ」およびヴィサルガ(Visarga)「:」に対応する翻字が定義されている．本研究では，これらの文字について，原綴り・翻字対応表を作成した．

また，単純な対応表では処理できない例外として以下のものがある．

- 鼻音化記号であり主に子音の前に来るアヌスワーラ(Anusvara)は，続く文字の種類に応じて6通りに翻字し分けなければならない．
- 鼻音化記号であり母音と子音に付くアヌナーシカ(チャンドラ・ビンドゥ)(Anunasika)は，続く文字の種類に応じて2通りに翻字し分けなければならない．
- 次の場合を除いて，全ての子音の後ろには母音「a」を付けなければならない．
  －他の母音が子音に続く場合．
  －母音を省略する特別記号である「ハル記号」(halant)が続く場合．

本研究では，これらの例外を処理する例外規則を作成した．なお，規則の動作確認のため，約600の原綴り・翻字データを作成し，繰り返し規則の修正を行なった．

---

5 現代のロシア語では使われない，あるいは，あまり使われない8文字も含まれる．

6 ウルドゥー語からの外来語で使われる文字も含まれている．

3.3. 翻字から原綴りへの自動変換

翻字規則は，原綴りを翻字に変換する点だけを考慮しており，翻字から原綴りへの変換については，定義された規則は存在しない．そのため，本研究では，翻字と原綴りを参照しながらALA-LC方式の翻字規則を逆に変換する規則の作成を行なっている．原綴りを翻字に変換する場合と同様に，翻字側の各文字を原綴りに対応させれば処理できる部分と，例外的な処理の必要な部分が存在する．そこで，基本的には原綴りから翻字への変換で作成した「原綴り・翻字対応表」を逆にして，「翻字・原綴り対応表」を作成する．そのうえで，対応表では処理できない部分については，個別に「例外規則」を作成する．

対応表と例外規則による翻字から原綴りへの自動変換の基本的な処理手続きは次のようになる．まず，翻字の先頭から１文字を取り出し，その文字を含む例外規則が存在するかどうかを調べる．もし，例外規則が存在する場合は，その翻字の前後（言語や例外の種類によりどれだけの範囲を見る必要があるかは異なる）の翻字も調べ，例外に該当するかどうかを判定する．例外に該当する場合は，その例外規則によって原綴り化を行なう．該当しない場合，および，例外規則に関連がない場合は，対応表に基づいて文字レベルでの原綴り化を行なう．

ロシア語

ロシア語では，１種類の翻字が複数のキリル文字に対応するということはない．そのため，翻字から原綴りへの変換も大部分は対応表で対処できる．ただし，異なる翻字間で同じ文字が重複するものがあるため，その点を処理する例外規則が必要になる．本研究では，７６種類の「翻字・原綴り対応表」を作成した．内５０種類については対応表だけで対処可能であるが，２６種類の翻字は，表2に示す文字の重複がある１０グループに分けられるので，例外規則を作成した．

また，Shch と shch に関しては，他の翻字「Sh」，「sh」および「ch」との間で以下の曖昧性が存在する．

- 「Shch」として「Щ」に変換するか，「Sh」と「ch」に分けて「Ш ч」に変換するか．
- 「shch」として「щ」に変換するか，「sh」と「ch」に分けて「ш ч」に変換するか．

この２点について，ロシア語話者に確認したところ，ロシア語では，Щと ч，あるいはщと ч が連続することはほとんどありえないため，それぞれЩとщであると考えて差し支えないという解答を得た．そのため，本研究では，Shch，shch を１つの単位として原綴りに変換することとした．

なお，規則の動作確認のため，原綴りから翻字への変換で用いたものと同様の約２００データを用いて，繰り返し規則の修正を行なった．

表２：　文字の重複がある翻字

翻字	原綴り（キリル文字）
Z, Zh	З，Ж
I, I͡E͡, I͡U͡, I͡A͡	И, Ь, Ю, Я
K, Kh	К，Х
S, Sh, Shch	С，Ш, Щ
T, T͡S͡	Т，Ц
z, zh	з，ж
i, i͡e͡, i͡u͡, i͡a͡	и, ь, ю, я
k, kh	к，х
s, sh, shch	с，ш，щ
t, t͡s͡	т，ц

ヒンディー語

ヒンディー語の翻字から原綴りへの変換は，ロシア語に比べて複雑である．基本的には，原綴り・翻字対応表を逆にした，翻字・原綴り対応表を作成するが，以下の例外について対応する必要がある．

- 異なる翻字内に同じ文字が使われているものが，母音８種類３グループ，子音３０種類１３グループあるため，これらについては場合分けをする必要がある．
- 母音・二重母音全１９種は，原綴りでは単独の場合と子音に接続する場合で文字が異なるが，翻字では区別がないため位置と他の翻字の並びから判断する必要がある．
- 子音の後に母音を表す翻字が存在しない場合，母音が省略されているので，原綴りにハル記号を加える必要がある．
- アヌスワーラを表す６種類の翻字の内，４種類「n, ṇ, ṅ, m」は子音「न, ण, ङ, म」の翻字と同じであるため，アヌスワーラであるか，子音であるかを区別する必要がある．翻字規則においてアヌスワーラは後に続く子音により翻字の種類が決まるので，翻字の並びからアヌスワーラになる可能性があるかどうかを判別できる．

ただし，最後の例外において，アヌスワーラで表せると判別された翻字の並びは，子音+ハル記号でも原綴り化できるという曖昧性が存在する．例えば，nandana という翻字の３文字目の「n」の扱いの違いで，「नंदन」と「नन्दन」とが考えられる．この点について，ヒンディー語の登録作業

者に確認したところ，意味的にはどちらも同じであるが，辞書等ではアヌスワーラを用いる方が一般的である，との解答を得た．そのため，本研究では，アヌスワーラとして変換することとした．

なお，規則の動作確認のため，原綴りから翻字への変換で用いたものと同様の約６００データを用いて，繰り返し規則の修正を行なった．

4. 実験

本システムの精度を確かめるため，ロシア語とヒンディー語について，未知のデータによる以下の２種類の実験を行なった．

実験１ 原綴りから ALA-LC 翻字への変換

実験２ ALA-LC 翻字から原綴りへの変換

4.1. 実験１

実験１では，原綴りから ALA-LC 翻字への変換の精度評価実験を行なう．

実験の評価用データとして，原綴りとその翻字の対を用意し，原綴りをシステムに入力する問題，翻字を正解として正解率を計算する．今回の実験では，東京外国語大学附属図書館のロシア語とヒンディー語の書誌情報の中からそれぞれ２８３，２７０のデータを無作為に抜き出して評価用データとする．東京外国語大学附属図書館では，非ローマ字母系言語の書誌情報を原綴りと翻字の両方で登録しているため，本実験にデータとして利用することができる．

各データは主に書誌タイトルの原綴りと翻字の対になっているため，分ち書きされた複数の単語を含んでいる．そのため，正解率の計算は１レコード全体でなく，分ち書きされた単語ごとに計算することとする．実験結果を表３に示す．

表４：実験結果　原綴りから翻字への変換

言語	件数	分かち数	正解	正解率
ロシア	283	1365	1363	99.9%
ヒンディー	270	2518	2502	99.4

結果から，今回の実験ではロシア語についてはほぼ正しく翻字に変換することができた．システムの出力が誤りだった２例は，単純な変換表の記述ミスであったため，すぐに修正が可能である．

ヒンディー語についても，ほぼ正しく翻字に変換することができた．特に今回のデータでは，原綴り・翻字対応表の範囲での誤りはなかった．１６例の誤りはすべて例外規則に関するものであり，子音の後ろに母音「a」を補う規則とアヌスワーラの翻字における場合分けの規則に不具合があることが原因だった．今後，この点を修正することで精度は向上するものと思われる．

4.2. 実験２

実験２では，ALA-LC 翻字から原綴りへの変換の精度評価実験を行なう．

実験の評価用データとして，実験１と同じデータを用いる．ただし実験２では翻字を，システムに入力する問題，原綴りを正解として正解率を計算する．また，正解率の計算は，実験１と同様に１レコード全体でなく，分ち書きされた単語ごとに計算することとする．実験結果を表５に示す．

表５：実験結果　翻字から原綴りへの変換

言語	件数	分かち数	正解	正解率
ロシア	283	1365	1334	97.7%
ヒンディー	270	2518	2294	91.1

結果から，今回の実験ではどちらの言語の場合も実験１に比べて正解率は低いものの，９０%以上の精度を得ることができた．

ロシア語については，かなりの精度で正しく原綴りに変換することができているといえる．今回のデータでは翻字・原綴り対応表に誤りはなく，システムの出力が誤りだった３１例は以下の原因によるものだった．

- 原綴りでъが単語の最後に来る場合は翻字されないという翻字規則により，翻字側に情報がないため，原綴りでъが再現できない誤り(２例)
- 翻字にまざっていた英語部分が翻字の字母と一致したため，キリル文字に変換された誤り(3 例)
- 翻字にまざっていた記号が翻字で用いる字母と一致したため，キリル文字に変換された誤り(２６例)

上記の内，１つ目の誤りは，元の翻字規則において非可逆な変換となるため，完全な自動化は不可能である．対策としては，ъが最後に付く単語を収集するなどして，ъが省略されている可能性の高い語が出現した場合に，利用者に確認を促すことなどが考えられる．他の２つの誤りは，キリル文字に他の文字や記号がまざることが原因であり，純粋なロシア語の問題ではないが，実際の書誌情報では頻発するため，間違いになり易いパターンを登録しておき，変換時に利用者に確認をするなどの対処をする必要がある．

ヒンディー語については，誤りが２２４例あっ

た．その原因は以下のようである．

- 母音の省略を示すハル記号が適切に挿入されていない誤り(１７例)
- 3.3.2 節で，アヌスワーラか子音+ハル記号とするか曖昧性がある場合，本研究ではアヌスワーラでの原綴りを行なうことにしたが，実際の図書では子音+ハル記号が用いられていた誤り(１９６例)
- 翻字にまざっていた記号が翻字で用いる字母と一致したため，デーヴァナーガリー文字に変換された誤り(１１例)

1つ目の誤りは，例外規則を見直す必要がある．2つ目の誤りは，アヌスワーラをデフォルトとして選択した部分でかなりの誤りがあるため，他の候補を利用者に提示することも検討する必要がある．3 つ目の誤りは，ロシア語の場合と同じく，利用者に確認をするなどの方法も検討する必要がある．

5. まとめと今後の課題

本稿では，我々が開発している非ローマ字母系言語の原綴り・翻字相互の自動変換システムについて述べた．最初の開発言語として，ロシア語とヒンディー語によるシステムを実装し，実験による評価を行なった．結果から比較的高精度で変換が行なえているといえる．本システムを更に良くするため，今後の課題として次のことがあげられる．

- 例外規則等の修正．　今回実装した 2 言語について，実験により明らかになった例外規則の問題点を修正し，より精度の高い相互変換を実現する必要がある．
- 他の言語への対応．　2.2 節で難易度を検討し，今回実装を見送った他の言語についても，今後モジュールを作成し，本システムで扱える言語を拡大していく必要がある．現在，我々は次の開発言語をアラビア語に定め開発にとりかかっている．

また，本システムによって提供される「原綴りから翻字への自動変換」と「翻字から原綴りへの自動変換」機能は，図 2 に示すような枠組で「書誌情報登録支援システム」や「蔵書検索システム」に応用されることにより効果を発揮する．

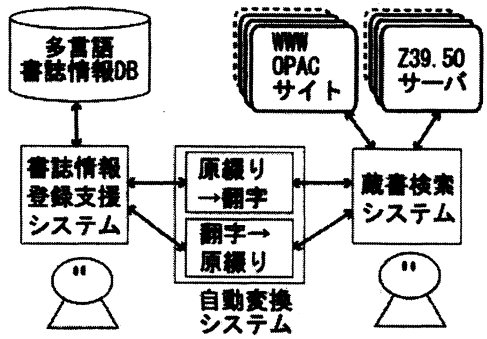

図 2：応用システムへの拡張

本研究の今後の課題として，現在我々は上記の応用システムの構築にも取りかかっている．書誌情報登録支援システムにおいては，翻字作成の作業を効率化するだけでなく，将来の書誌情報の本格的な多言語化を見据えて，既存の書誌情報内の翻字から原綴りを作成する支援も視野に入れている．また，検索システムについては，世界中の ALA-LC 方式の翻字を用いる図書館サイトの中で，インターネットを系由した Z３９．５０や HTTP-OPAC での接続が可能なサイトに対する横断検索の実現を予定している．

参考文献

[1] G.M.McCune and E.O.Reischauer. The Romanization of the Korean Language. In *Transaction of the Korea Branch of the Royal Asiatic Society* 29, pp.1-55, 1939.

[2] Unicode Inc. The UnicodeCharacter Code Charts. http://www.unicode.org/charts/

[3] R.K.Barry, editor. *ALA-LC Romanization Tables -Transliteration Schemes for Non-Roman Scripts-*. Library of Congress, 1997.

[4] 国立情報学研究所．NACSIS-CAT/ILL:目録所在情報サービスホームページ．http://www.nii.ac.jp/CAT-ILL/contents/home.html

[5] 中嶋仁．現代朝鮮語の言語規範-その変遷と認識度調査を中心に-，東京外国語大学 語学研究所論集，Vol.7，pp.119-144，2002.

[6] 林哲也．技術的知識としての実用語学：翻字とタイ語早見表を例として．大学図書館研究，Vol.44，pp.40-49，1994.