F-03

対訳コーパスに基づくローマ字化システムの構築時における必要学習コーパス量の推定

Estimation of Necessary Data size for Inducing Romanization System from bilingual Corpus

田口 恵子†　フィンチ アンドリュー‡　山本 誠一†　隅田 英一郎‡
Keiko Taguchi　Andrew Finch　Seiichi Yamamoto　Eiichiro Sumita

概要

本稿は対訳コーパスから書記素単位でローマ字化システムを構築する手法についての実験・分析結果を述べる．提案手法をヒンディー語に適用しトランスリテレーションマイニングタスクにおけるマイニング性能の比較実験を行った．さらに日本語，ロシア語，ヒンディー語，ミャンマー語に提案手法を適用し言語によるローマ字化システム構築の難易度の違いを言語の曖昧性の観点から分析した．また提案手法によるローマ字化システムの十分な学習に必要なコーパスサイズを推定する．まず適切なローマ字化システムの構築に必要な書記素の発生頻度を求め，べき乗則によって未観測の書記素の発生確率を推測し学習コーパス量を推定する．

1. はじめに

筆者らは先行研究[1]で日英対訳コーパスから書記素ごとにローマ字化システムを統計的に構築する手法を提案した．提案手法はあらゆる言語に適用可能な対訳単語ペアに基づくローマ字化システムを構築する新しい手法であり，既存のローマ字化システムとは異なりデータマイニングやユーザーインプットなどの目的に適応したローマ字化システムを構築することができる．構築したローマ字化システムをトランスリテレーションマイニングタスクに利用したところマイニング性能の向上が見られた．しかしローマ字化システムの学習能力は原言語の書記素数に対するデータサイズに大きく影響されることが分かった．

そこで新たに多様な既存のローマ字化システムを持つヒンディー語と標準のローマ字化システムを持たないミャンマー語において提案手法を適用し計 4 つの全く違う言語（日本語，ロシア語，ヒンディー語，ミャンマー語）の言語の曖昧性から言語によるローマ字化システム構築の難易度の違いを分析し，また提案手法の実用性を示すためにローマ字化システムの学習が難しい書記素数が多くかつコーパス資源の少ない言語であるミャンマー語においてローマ字化システムの十分な学習に必要なコーパスサイズを推定する．

2．ローマ字化システム

ローマ字化とは，ラテン文字以外の文字体系を持つ言語をラテン文字（ローマ字）によって表記することである．言語によっては他の言語には存在しない特有の発音や音の区別があり，一般的には音から文字を表記する転写だけでローマ字化を行うことは不可能でありローマ字化を行う場合，音の表現に言語間での偏りが発生する．

例えば，日本語では 2 つの主要なローマ字化システム：ヘボン式ローマ字・日本式ローマ字が存在する．前者は音素転写を用いて出来るだけ英語の音素に忠実に再現するように設計されている．後者は固有の文字体系であるかな文字をローマ字に対応付けしてローマ字化を行っておりローマ字化後のローマ字の文字列が英語のように発音されることよりも元々のかな文字の発音形態を保つことに重きが置かれている．

2.1　ローマ字化システム構築の意義

一見，日常生活で既に確立されているローマ字化システムを構築する試みは奇妙に思えるが，近年ローマ字化システムは既存のローマ字化システムが設計された当初には存在しなかったトランスリテレーションマイニングやテキスト入力などの新しい役割を担うようになってきており，より目的に適した最適なローマ字化システムの開発が期待されている．

トランスリテレーションマイニングでは原言語と目的言語の文字体系の統一のためにローマ字化システムが使用されている[2, 3]．またテキスト入力では多くの言語ではそれぞれの文字体系全てを直接ユーザーインターフェース上で表現するには文字の種類が多すぎるため一般的に既存のローマ字化システムを用いてキーボード上でそれぞれの文字体系をローマ字の文字列として表現している．その一例として中国語のテキスト入力ではピンインが用いられている．しかし第一に問題となるのは既存のローマ字化システムは元々テキスト入力での使用を想定しておらず，一文字一文字のかすかな発音の違いなど書記素がどのように発音されるべきか明確に述べるために入力が煩わしい長々と続く表現が多く存在している．

更にローマ字化システムを新たに構築する他の理由として次のようなことが挙げられる．ある言語に対して競合するローマ字化システムが複数，存在することがしばしばある．ユーザーはテキスト入力に複数あるローマ字化システムの内のひとつを使用するか，それらを混合したローマ字化システムを使用している．日本語でもテキスト入力に用いるローマ字化システムは一意的ではなく，主としてユーザーによってテキスト入力に使用するローマ字の入力作法が異なる．例えば日本語の“ち”を入力する場合，表現方法は‘chi’（ヘボン式ローマ字）‘ti’（日本式ローマ字）があり‘ti’の方が表記が短いために好んで使用されるが‘chi’の方がより正確に文字の発音を反映している．

入力効率と発音表現の正確性点で日本ではテキスト入力として使用される既存のローマ字化システムはどちらもユーザーインプットとして完璧ではなく未発見のより良いローマ字化システムが存在すると考える．ユーザーイ

‡ 独立行政法人情報通信研究機構，NICT
† 同志社大学，Doshisha University of Information Engineering

ンプットに適したローマ字化システムを発見するタスクにおいて考慮すべき点が 3 つある；①ローマ字化した文字の発音がどのぐらい元の文字の発音を表現しているか，②入力は効率的か，③ローマ字化に曖昧性が発生しないかの 3 点である．

本稿では言語間の単語の類似性におけるローマ字化の問題に対して取り組む．ローマ字化システムの構築時に複数あるローマ字化候補の中からひとつローマ字を選択する段階で選択基準がシンプルで明確であり，また構築したシステムの性能の評価や分析が単純明快であるからである．しかしながら私たちはこの技術がより広く適用され，提案手法がより複雑な基準によって構築されたローマ字化システムが他の目的で使用され発展していくと考えている．

提案手法の主なメリットはある目的に特化したローマ字化システムの構築が可能な点，学習コーパスさえ存在すれば標準のローマ字化システムやそもそもローマ字化システムが存在しない言語(ミャンマー語[4])などあらゆる言語に適用できる点，また既存のシステムの代わりとなるローマ字化システムを構築できる点である．私たちは以下で日本語やロシア語[1]に限らずヒンディー語においてもトランスリテレーションマイニングタスクで十分に確立された既存のシステムと同等かそれ以上の効果を構築したローマ字化システムがもたらすことを示す．

2.2 提案手法

まず対訳コーパス(原言語 - 英語)からベイジアンアライメント[5]を用いて書記素ごとに英語の対応付けを行い，ローマ字化変換ルール候補を得る．ベイジアンアライメントの特徴は過学習なしで一貫性のあるアライメントが可能である点である．書記素とローマ字の対応付けにおいては一対多，多対多のアライメントを取る．その変換ルール候補の中からローマ字化システムの使用目的に適した基準用いてその目的に最も適切なローマ字化変換ルールを一意的に決定する．本稿ではトランスリテレーションマイニングタスクでの適用を目的とするので多くの

トランスリテレーションマイニング[6, 7]で一般に使用される標準化編集距離(Normalized Edit Distance: NED)を応用した期待編集距離(Expected Edit Distance: EED)を基準に採用した．

対訳コーパスの原言語と目的言語(英語)を $S = (s_1, s_2, \ldots, s_I)$ と $T = (t_1, t_2, \ldots, t_I)$ とする．各 s_i と t_i はそれぞれの文字体系における書記素とする．

Π と Ω はそれぞれの文字体系における書記素とアルファベットの集合である．ローマ字化変換ルール R はタプル (o_j, r_j) の集合である．o_j と r_j は原言語の書記素と英語のローマ字である：$\forall j\ o_j \in \Pi,\ r_j \in \Omega$

$$R = \{(o_1, r_1), (o_2, r_2), \ldots, (o_J, r_J)\} \qquad (1)$$

$\Phi : \Pi \rightarrow \Omega$ は R によって定義されるローマ字化関数とする．r_j はベイジアンアライメントによって o_j とアライメントがとられたローマ字化ルール候補集合 C_j から選択する：$C_j = (c_1, c_2, \ldots, c_K)$．変換ルール (o_j, r_j) は式(2)に表す EED が最小値をとる r_j を選択する．

$$\phi(o_j) = \underset{c_k \in C_j}{\operatorname{argmin}}\, E[D(c_k)] \qquad (2)$$

$D(c_k)$ は変換ルール候補 (o_j, c_k) のレーベンシュタイン距離のコストを表す．対訳コーパス中の o_j に対して候補が 1 つしかない場合は，そのコストは変換ルール候補 c_k と o_j と対応付けされたローマ字 $\psi(o_j)$ とのレーベンシュタイン距離 $LD(c_k, \psi(o_j))$ とする．

対訳コーパスにおけるこのコストの期待値は以下のように計算される．

$$E[D(c_k)] = \sum_{l=1..K} p(c_l) LD(c_k, c_l) \qquad (3)$$

レーベンシュタイン距離のコストの期待値，つまり発生確率とレーベンシュタイン距離を組み合わせた EED が最も小さくなる変換ルールを選択することで生成されるローマ字に原言語と目的言語の偏りをなるべく抑えてローマ字化を行う．

3. トランスリテレーションマイニング

評価実験として[6, 7, 8]と同様に NED を元に対訳単語ペアをトランスリテレーションペアであるか否か判別するトランスリテレーションマイニングを行う．NED はローマ字化した単語と英単語との間で計測し分類を行った．ここでは NED はレーベンシュタイン距離を編集経路の和で割ったものである．編集経路の和で割ることで測定する 2 つ文字列の長さの違いによる値の偏りを除去することができ NED の値域を[0,1] に留めることができる．

本研究で用いる言語はヒンディー語である．ヒンディー語はデーヴァナーガリー文字を文字体系として持つ言語であり少なくとも 7 つのローマ字化システムが存在する．本研究の比較対象として京都・ハーバード方式のローマ字化システムを選択した．京都・ハーバード方式はヒンディー語においてパソコン通信などで用いられる一般的なローマ字化システムである．ヒンディー語のローマ字化システムを構築するために NEWS2010 Shared Mining Task[9]のウィキペディアの他言語への内部リンクから抽出された対訳コーパスから単一単語対の対訳単語ペアのみを使用する．学習データは NEWS2010 のトレーニングデータから抽出した 3736 単語ペア，テストデータは NEWS2010 のシードデータ 1000 単語ペアとトレーニングデータから抽出した 246 単語の計 1246 単語ペアである．

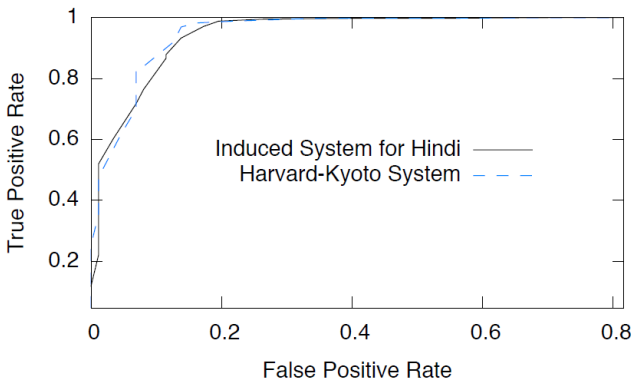

図 1　ヒンディー語の ROC 曲線

トランスリテレーションマイニングタスクにおいて多対対アライメントで学習・構築したローマ字化システムと京都・ハーバード方式のローマ字化システムの性能を比較する．NED を閾値として各分類器の性能を表す ROC 曲線(Receiver Operating Characteristic curves)を図 1 に示す．ROC 曲線は弁別閾値を変化させて二項分類器の性能を示すグラフである．

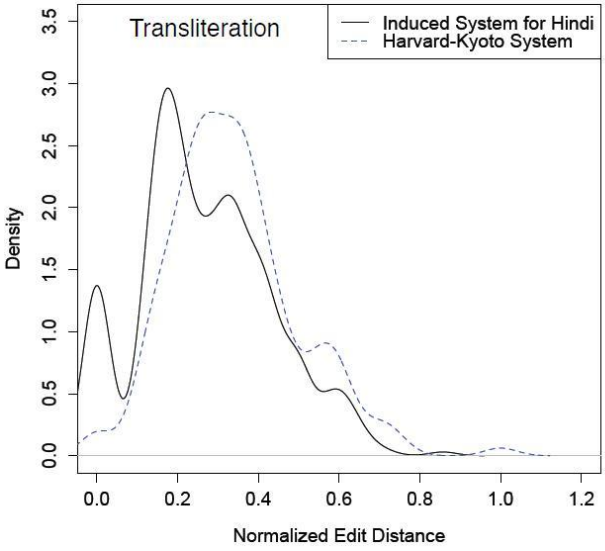

図 2　トランスリテレーションペアの NED 分布

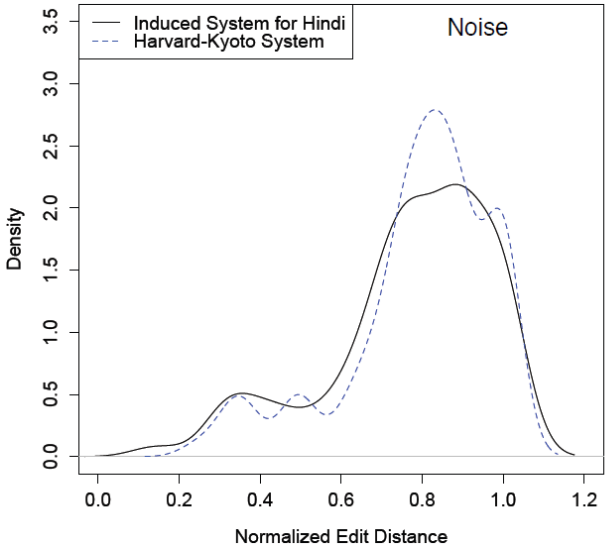

図 3　ノイズペアの NED 分布

図 1 は提案システムと京都・ハーバード方式はほぼ同等の分類性能を示している．図 2，3 はテストデータ中のトランスリテレーションペアとノイズペアのそれぞれの NED の確率密度関数(Probability Density functions: PDF)を求めカーネル密度推定を行ったグラフである．提案手法はトランスリテレーションペアに対してのみ編集距離を減少させ両分布の境界を明確にし，分類性能を向上させる戦略であるがヒンディー語ではトランスリテレーションペアの編集距離を減少させるが，ノイズペアに対しても僅かに編集距離を低下させている．その結果，分類性能が既存のローマ字化システムと同等の性能を示したと考えられる．

京都・ハーバード方式のローマ字を分析したところ，その半数以上(36/51 書記素)がローマ字と一対一の置換でローマ字化が行われていることが分かった．一対一のシンプルな置換と分類性能結果からヒンディー語と英語は系統的な語族は大きく違うが，提案手法によるローマ字化システムと京都・ハーバード方式のローマ字化システムは類似していると考えられる．

4. 言語の曖昧性の分析

先行研究[1]で適用した日本語，ロシア語と本研究で新たに適用するヒンディー語とミャンマー語のパープレキシティ(perplexity)を利用して言語の曖昧性から言語によるローマ字化システム構築の難易度の違いを分析する．

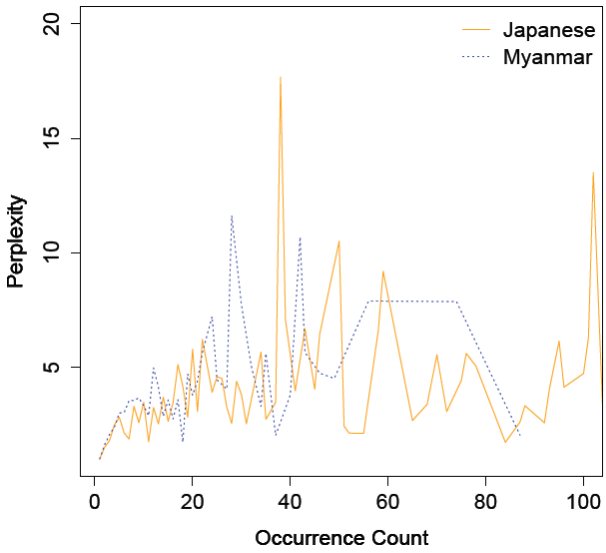

図 4　書記素の発生回数に対する言語の曖昧性

図 4 は日本語とミャンマー語のそれぞれの書記素の発生回数に対するパープレキシティを示す．図 4 のデータはノイズが多いが，日本語とミャンマー語の書記素の曖昧性はおおよそ同等である．全コーパスに対する平均のパープレキシティは日本語とミャンマー語は共に 4.37 である．発生回数が 1 回の書記素はパープレキシティの過小評価を招くため平均算出時にそれらの書記素を除くとパープレキシティは日本語では 4.41，ミャンマー語では 4.46 となる．この結果はミャンマー語のローマ字化処理はほとんど日本語のローマ字化処理と同等の難易度であることを示唆している．

ロシア語のパープレキシティは 1.76 でありロシア語のキリル文字とローマ字とのアライメントに曖昧性がほぼないため日本語やミャンマー語よりも大幅にロシア語のローマ字化処理は容易であると考えられる．

またヒンディー語のパープレキシティは 8.03(発生回数 1 回の書記素を除いた場合 8.14)である．デーヴァナーガリー文字はローマ字とおおよそ一対一でアライメントが取られるので一見ローマ字化処理が容易であると考えられるが，アライメントの段階で 1 つの書記素に対して多

数のローマ字が対応付けされる．それゆえ発生回数が少ない書記素においても沢山のローマ字化候補があるためにパープレキシティの値が押し上げられたと考えられる．

5. 必要学習コーパス量の推定

提案手法によるローマ字化システムの性能はコーパス中に含まれる書記素とローマ字のアライメント数に依存する．学習データ量は提案手法において重要な要素であり，提案手法には書記素数が多い文字体系を持つ言語は学習により多くのコーパス量が必要になるという潜在的な問題がある．そこでローマ字化システム構築に要する書記素数に対した学習コーパス量を推定する．

5.1 学習データ量のシステム性能への影響

まず予備実験として学習データ量のローマ字化システム性能への影響についてヒンディー語と同様に NEWS2010 [9]の日本語の学習データを用いて実験を行う．

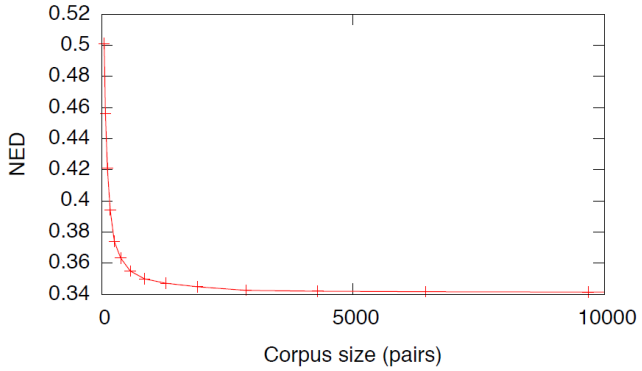

図 5　学習データの変動による NED の遷移グラフ

図 5 のそれぞれの点は学習データからランダムサンプリングで各コーパス量において 100 回ローマ字化システムを構築し平均 NED を計測した結果である．NED は約 3000-4000 単語ペアでの学習で値が収束している．

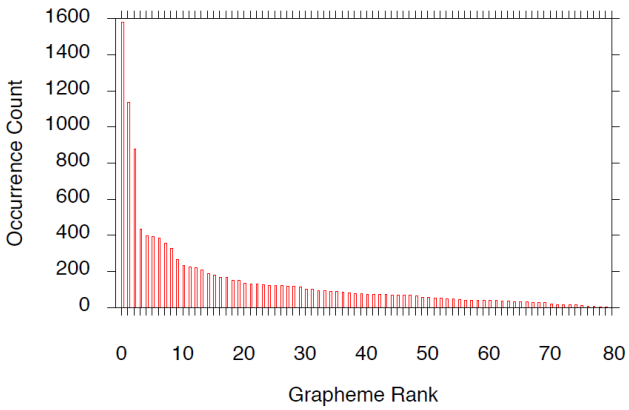

図 6　書記素の発生回数の分布

図 6 は 3200 組の対訳単語ペア中に含まれる書記素の発生回数の分布を示した図である．図 6 は典型的なロングテ-ル型の分布を成しておりグラフの左側の書記素が発生頻度が高いことを示している．また多くの書記素はコーパス中に数百回発生していることから必要な学習データ量とはつまり発生頻度の少ない書記素が十分に学習できるデータ量ということになる．

この実験では最も発生頻度の高い書記素の発生回数は 1583 回，最も少ない発生回数は 1 回，平均発生回数は 146 回である．これは 1 書記素が平均約 42 単語ペア中に発生することを意味する．

5.2 書記素の必要発生頻度の推定

十分な学習に必要な書記素の発生回数（アライメント数）を検証するために予備実験から 4000 単語以上の単語ペアで十分な学習が得られると仮定し[10]の日英対訳単語ペア 4339 組を用いて学習したローマ字化システムのローマ字化変換ルールを適切なローマ字化と見なしそれらのローマ字化変換ルールを獲得する確率を相対的発生頻度をベースとした書記素の確率分布から算出することで必要な書記素の発生回数を求める．

仮に書記素がコーパス中に唯一のアライメントしか存在しない場合でもルールの選択段階で書記素とローマ字の対応を取ることができるため適切なローマ字化を行うことは十分可能である．

適切なローマ字化変換ルールを獲得する確率はコーパス中に「適切なローマ字が含まれているアライメントが含まれているかどうか」のベルヌーイ試行であるので適切なローマ字を全く含まないアライメントを取る確率の余事象として計算される．

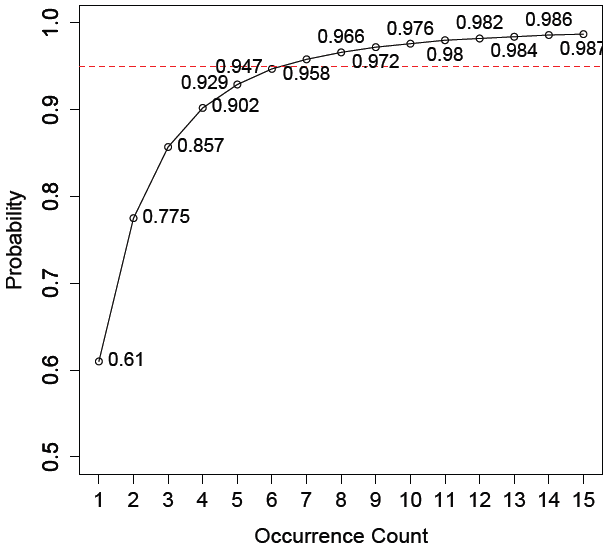

図 7　書記素の発生頻度推定

図 7 は書記素の発生回数に対する適切なローマ字化変換ルールの獲得確率を示している．x 軸は試行回数つまりコーパス中の書記素の発生回数を表し，y 軸は x 回書記素が発生したときにその書記素に対する適切なローマ字化変換ルールの獲得確率の全書記素に対する平均を表す．このグラフから十分な学習によるローマ字化システム構築成功の基準を 95%（グラフ中の赤破線）とすると書記素に必要な発生回数は 6, 7 回である．

5.3 必要コーパス量の推定手順

この項目ではミャンマー語を用いて適切なローマ字化システムの学習に必要なコーパス量について研究する．ミャンマー語は約 1800 種類の書記素がありコーパス資源が非常に乏しい言語である．現在のところ標準のローマ字化システムを持たず[Oo2009]，私たちの研究が役立てられる言語のひとつであると考えられる．ここではミャンマー語の書記素数を 1800 種類と仮定する．

私たちは日本語での適切なローマ字化に必要な書記素の発生回数についての分析結果を用いて必要なコーパス量の推定を試みる．以下の手順でコーパス量の推定を行う．

1. 適切なローマ字化システムの獲得に必要な書記素の発生回数の推定
2. 書記素に対する確率分布のべき乗分布への適合
3. 発生頻度の少ない書記素の十分な発生回数を達成するコーパス量の推定

手順 1 では第 4 節から日本語とミャンマー語の曖昧性がほぼ同等であるのでミャンマー語における書記素の必要発生回数は日本語の必要発生回数と同じであると仮定する．

5.3.1 べき乗分布への適合

手順 2 ではべき乗分布を実際のデータに適合することによって実データでは未観測である書記素の発生確率を推定する．出現頻度が k 番目に大きい要素が全体に占める割合が 1/k に比例するという経験則であるジップの法則を用いて書記素の発生頻度を表す．

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \qquad (4)$$

k は書記素の発生頻度の順位，n はコーパス中の全書記素数，s は分布の指数特性を表す．

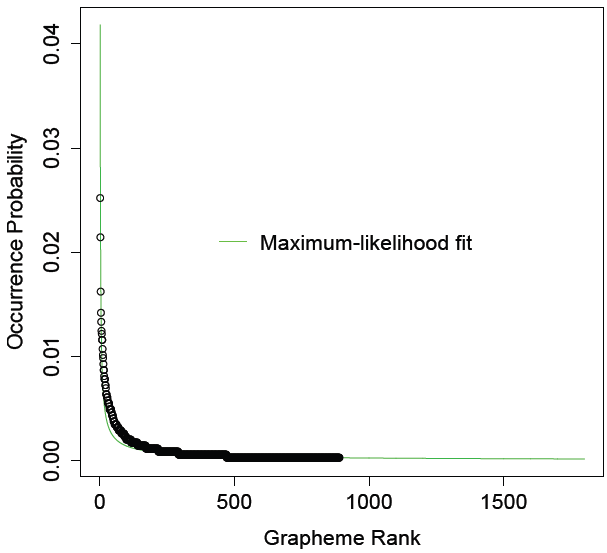

図 8　ジップ分布への適合

ジップ分布を適合した実データは 1076 組のミャンマー語と英語のトランスリテレーション単語ペアである．そのコーパスにはミャンマー語の書記素 1800 種中 890 種類の書記素が含まれる． 図 8 は実データに最尤推定（s = 0.74）を行いジップ分布に適合した結果である．グラフ中の黒点が実データを示す．

5.3.2 必要コーパス量の推定

手順 3 では適切なローマ字化に必要な発生回数の推定結果と未観測の書記素の発生確率を利用して適切なローマ字化に必要なコーパス量を推定する．

図 9 は全順位に置けるミャンマー語の書記素の期待発生回数とコーパスサイズの関係を示す．曲線はそれぞれ期待発生回数 6, 7, 8 回を表す．これは第 5.2 節の日本語での適切なローマ字化処理を得るために必要な書記素の発生回数は 6, 7 回という結果に由来している．図 9 から少なくとも 30,000 単語のコーパスが必要なことからミャンマー語の全書記素をローマ字化するのは現在のコーパス資源では実用的ではないことは明らかである．30,000 単語のコーパスは豊富なコーパス資源をもついくつかの言語のみに存在するサイズである．しかしながら図 9 の推定結果はとてつもなく膨大な量の対訳単語ペアではなく書記素数の多いミャンマー語において信頼できるローマ字化システムを構築する可能性を示唆している．

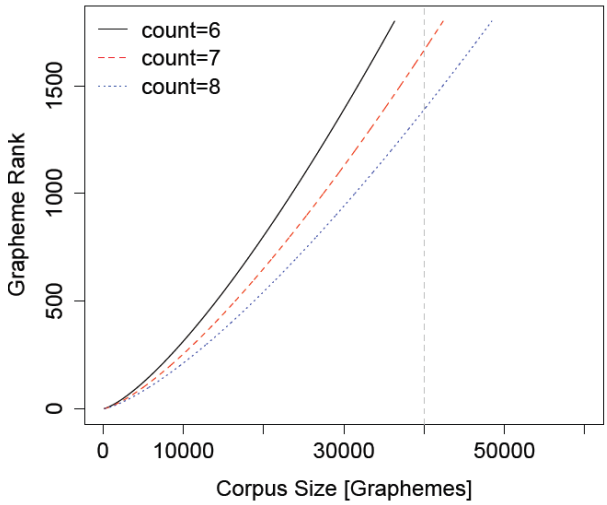

図 9　書記素数に対する必要学習コーパスの関係

6. まとめ

本稿は対訳コーパスから書記素単位でローマ字化システムを構築する手法についての実験・分析結果を述べた．提案手法をヒンディー語に適用しトランスリテレーションマイニングタスクにおけるマイニング性能の比較実験を行い，同程度の性能が得られた．

さらに日本語，ロシア語，ヒンディー語，ミャンマー語において言語によるローマ字化システム構築の難易度の違いを言語の曖昧性の観点から分析し，日本語とミャンマー語のシステム構築の難易度がほぼ同等であることを示した．またロシア語については言語の曖昧性がほとんどなくローマ字との対応付けがおおよそ一対一で行われることからロシア語のローマ字化システムの構築は日本語やミャンマー語に比べて容易であると考えられる．ヒンディー語においては一対一の対応にも関わらず高いパープレキシティ値を取り，これはヒンディー語では発生

回数が少ない書記素においても沢山のローマ字化候補があるためにパープレキシティの値が押し上げられたと考えられる．

また提案手法の実用性を示すためにローマ字化システムの学習が難しい書記素数が多くかつコーパス資源の少ない言語であるミャンマー語においてローマ字化システムの十分な学習に必要なコーパスサイズを推定した．まず同等の言語の曖昧性をもつ日本語のデータから適切なローマ字化システムの構築に必要な書記素の発生頻度を求め，ミャンマー語のコーパスをべき乗則によって適合することで未観測の書記素の発生確率を推測し学習コーパス量を推定した．推定結果から現在のミャンマー語のコーパス資源では実用的なローマ字化システムを構築することは難しいが，果てしない膨大な量の対訳単語コーパス量ではないことから将来的に書記素数の多いミャンマー語において信頼できるローマ字化システムを構築する可能性を示唆している．

今後の課題として，私たちは異なる基準を用いてローマ字化システムを開発し，ローマ字化システムに与える影響を具体的に研究する予定である．特に標準化システムを持たない言語への貢献を目指し言語の偏りのないよりハンドメイドに近いローマ字化システムの構築について研究する．私たちはこの技術が発展し基本的な書記素の充分な発音表現と入力効率を両立するテキスト入力に適した新たなローマ字化システム発見に繋がることを目指している．

参考文献

[1] 田口惠子, Finch Andrew, 山本誠一, 隅田英一郎. “対訳コーパスに基づく最適なローマ字化システムの構築” 第 12 回情報科学技術フォーラム(FIT2013), pp.111-116, 2013.

[2] K. Knight and J. Graehl. “Machine transliteration” Computational Linguistics, pp. 599-612, 1998.

[3] Y. Qin. “Phoneme strings based machine transliteration” Natural Language Processing and Knowledge Engineering (NLP-KE) on 2011 7th International Conference, pp. 304-309, 2011.

[4] N. P. Oo and N. L. Thein. “itextmm: Intelligent text input system for myanmar language on android smartphone” in IT Convergence and Services, pp. 661 - 670, Springer, 2011.

[5] A. Finch and E. Sumita. “A Bayesian Model of Bilingual Segmentation for Transliteration”, In M. Federico, I.Lane, M. Paul, and F. Yvon, editors, Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT), pp.259-266, 2010.

[6] W. Aransa, H. Schwenk, L. Barrault, and F. Le Mans. “Semisupervised transliteration mining from parallel and comparable corpora” Proceedings IWSLT 2012, 2012.

[7] O. Htun, A. Finch, E. Sumita, and Y. Mikami. “Improving transliteration mining by integrating expert knowledge with statistical approaches” International Journal of Computer Applications, vol.57, November 2012.

[8] S. Jiampojamarn, K. Dwyer, S. Bergsma, A. Bhargava, Q. Dou, M.Y. Kim, and G. Kondrak. ”Transliteration generation and mining with limited training resources” Proceedings of the 2010 Named Entities Workshop, pp.39-47, Association for Computational Linguistics, 2010.

[9] A. Kumaran, M.M. Khapra, and H. Li, “Report of news 2010 transliteration mining shared task” Proceedings of the 2010 Named Entities Workshop, pp.21-28, Association for Computational Linguistics, 2010.

[10] T. Fukunishi, A. Finch, S. Yamamoto, and E. Sumita. “Using features from a bilingual alignment model in transliteration mining”, In 2011 Named Entities Workshop, pp.49, 2011

[11] H.M. Oo, P.Y. Mon, K.T. Nakahira, and Y. Mikami, “Romanized myanmar input method for mobile phone” proceedings of the Seventh International Conference on Computer Applications, Yangon, Myanmar, pp.233-237, 2009.