混合ディリクレ分布を用いた潜在クラス翻字生成モデル

萩原 正人　　関根 聡
楽天株式会社 楽天技術研究所
{masato.hagiwara, satoshi.b.sekine}@mail.rakuten.com

1　はじめに

表記体系の異なる言語間の音韻的な翻訳である翻字は，外来語を取り入れる主要な方法である．近年，発音を介さずに綴りの対応関係を統計的に学習し，翻字を検出・生成する手法 [1, 5] が主流になりつつある．しかし，外来語には異なる起源を持つ語が混在するため，単一の翻字モデルでは関係を捉えきれないという問題が生じる．例えば，“piaget ／ピアジェ”(仏) と “target ／ターゲット”(英) の “get” のように，原言語が異なる場合，音韻・綴りの対応も異なる場合がある．

この問題に対して，言語・性別等の原言語の起源を明示的にモデル化し，翻字モデルを切り替えて用いる**クラス翻字モデル**が提案されている [6]．この手法では原言語の起源をタグ付けした学習データが必要であるが，このようなデータを入手することは難しい．この問題に対して我々は，原言語の起源を，直接観察できない潜在クラスとしてモデル化し，翻字ペアに対して尤もらしい翻字モデルを適用する**潜在クラス翻字モデル** (LST; Latent Semantic Transliteration) を提案した [2]．これによって，例えば “piaget ／ピアジェ” と “target ／ターゲット” にはそれぞれフランス語，英語に対応する潜在クラスが結びつき，翻字関係を正しく認識できると期待される．しかしながらこの LST モデルは，多項分布の最尤推定に基づいているため，変則的な発音を持つ翻字ペアのような学習データ中のノイズに弱く，過学習してしまうという問題がある．

ここで，“get ／ゲット” などのような最小書き換え単位 (**翻字単位**) を単語，“piaget ／ピアジェ” などのような**翻字ペア**を単語の列から構成される文書とみなすと，翻字は文書のトピックモデルにより定式化できる．実際 LST モデルでは，翻字単位の多項分布の混合によって翻字確率を定義しており，これはトピックモデルの一種であるユニグラム混合 (Unigram Mixture; UM) [7] の変種として定式化が可能である．ユニグラム混合に対して，混合ディリクレ分布による事前分布を導入し，過学習の問題を抑えたトピックモデルが提案されており [8]，同様の枠組みを用いて翻字モデルを定式化することにより，さらに高精度な翻字モデルが構築できると期待できる．

そこで本研究では，原言語の起源に基づく翻字モデルをトピックモデルの枠組みにより定式化し，混合ディリクレ分布に基づく潜在クラス翻字モデル (DM-LST) を提案する．評価実験により，提案手法が過学習の問題を抑え，優位な翻字性能を示すことを明らかにする．なお，本研究では，**翻字生成**のタスクを扱う．翻字生成とは，語 $\boldsymbol{s}$ (例えば “piaget”) が与えられた時に，その翻字先として最も適切な翻字先 $\boldsymbol{t}$ (例えば “ピアジェ”) を生成する．一方，翻字のタスクとして**翻字検出**があり，これは，入力語 $\boldsymbol{s}$ と出力候補の集合 T が与えられた時に，その中から翻字先として最も適切な $\boldsymbol{t} \in T$ を**翻字モデル** $P(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ を用いて $\boldsymbol{t}^* = \arg\max_{\boldsymbol{t} \in T} P(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ として求めるタスクである．

本稿では，まず2節にて，綴りに基づく翻字モデルのうち，最も基本的なアルファベータ法 [1] を紹介する．続く3節において，Joint Source Channel (JSC) モデル [5] を紹介し，アルファベータ法との関係について述べる．4節では，その拡張として LST モデル [2] について述べ，5節において提案手法である DM-LST を提案する．6節では評価実験の結果を示す．

2　アルファベータ法

翻字ペアの綴りの書き換え確率を直接モデル化する翻字モデルのうち最も単純なものとしてアルファベータ法 [1] について述べる．アルファベータ法は，文字の置換・挿入・削除それぞれの編集操作のコストを 1 とみなす通常の編集距離の一般化であり，$s_i \to t_i$ (s_i, t_i は長さ 0 以上, w 以下の文字列) の形の文字列書き換え操作に対して確率値を与える．この書き換え操作の単位 $u_i = \langle s_i, t_i \rangle$ を，文献 [5] にならい本稿では翻字単位と呼ぶ．本モデルを用いて，語 $\boldsymbol{s}$ を語 $\boldsymbol{t}$ に書き換える確率は，以下のように求められる：

$$P_{AB}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle) = \max_{u_1 \ldots u_f} \prod_{i=1}^{f} P(u_i) \tag{1}$$

ただし，$u_1 \ldots u_f$ は，入力語・出力語の翻字ペア $\langle \boldsymbol{s}, \boldsymbol{t} \rangle$ を分割してできる任意の翻字単位系列であり，例えば “pi ／ピ a ／ア get ／ジェ” である．上式は，翻字単位確率を独立と仮定しその積により翻字ペアの確率を定義したとき，確率を最大にする分割 $u_1 \ldots u_f$ を見つける問題に相当する．全体の対数を取り，$-\log P(u_i)$ を文字列書き換え操作 $s_i \to t_i$ のコストと見なすと，この問題は書き換えコストの合計の最小値を求める問題と等価である．よって，通常の編集距離と同様に動的計画法により解くことができる．

本モデルにおける翻字単位確率 $P(u_i)$ は，翻字ペアからなる学習データから学習する．しかしながら，学習コーパスには，$\boldsymbol{s}$ のどの文字が $\boldsymbol{t}$ のどの文字に対応するかという**アラインメント**の情報が無いため，文献 [2] では，文字列 s_i, t_i は同じアルファベット体系を使用すると仮定し，日本語のカタカナをローマ字表記に変換した．しかし本稿では，アルファベット体系の異なる文字列 (例

えば日本語カタカナと英語アルファベット，中国語漢字と英語アルファベット等)を直接対応付けられるように，そのような仮定を置いていない．そのため，文献 [1] にあるような，同じ文字の対応付けを用いてアラインメントを求めるアルゴリズムが使用できない．

そのため，本研究では翻字ペア間の可能な全てのアラインメントを考慮し，それらの相対頻度を数え上げることにより，翻字単位確率を計算した[1]．なお，本稿において扱う全てのモデルには，翻字単位として考慮する文字 n-gram の最大長 w というパラメータがあるが，本研究では一貫して $w = 3$ を用いた．

3 Joint Source Channel モデル

上記のアルファベータ法では，翻字単位のアラインメントにヒューリスティックを用いていること，および，翻字単位の依存関係[2]を捉えられないという問題がある．本節では，このアルファベータ法とは独立に提案された，翻字検出モデルである JSC モデル [5] について述べる．JSC モデルは，1) 翻字単位のバイグラム以上も考慮できる 2) 翻字単位の統計を取る際に，アラインメントをヒューリスティックにより固定するのではなく，EM アルゴリズム的な手法により逐次的に更新するという2点以外は，アルファベータ法と本質的に同じである．

JSC モデルでは，翻字単位 $u_i = \langle s_i, t_i \rangle$ の n-gram 確率を用いて以下のように翻字確率を計算する：

$$P_{JSC}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle) = \prod_{i=1}^{f} P(u_i | u_{i-n+1}, ..., u_{i-1}) \qquad (2)$$

ここで，f は式 (1) 同様，翻字単位の数である．翻字単位確率 $P(u_i|u_{i-n+1}, ..., u_{i-1})$ は，以下の EM アルゴリズム的な逐次更新により求められる：

1. 初期アラインメントをランダムに設定する．
2. E ステップ：現在のアラインメントを用い翻字 n-gram 統計を求め，翻字モデルを更新する．
3. M ステップ：現在の翻字モデルを用いアラインメントを更新する．アラインメントはアルファベータ法と同様の動的計画法により求められる．
4. 収束するまで 2. と 3. を繰り返す．[3]．

以上のアルファベータ法および JSC モデルは翻字検出モデルであるが，入力 $\boldsymbol{s}$ のみが与えられた時に，その翻字 $\boldsymbol{t}$ を出力するために，スタックデコーダを用いて確率の高い翻字候補を生成した．図 1にその概要を示す．入力として語 $\boldsymbol{s}$ (ここでは “smith”) が1文字ずつ与えられ，それが各候補の末尾の翻字単位に追加される (図中の「append」)．その後，各翻字単位を reduce もしくは shift する．reduce した場合，翻字単位の表 (図左下) を参照しながら，確率の高い上位 R 個の翻字単位を生成・確定する．これにより，図中 “s” が入力された後，(“s”, “ス”) および (“s”, “ズ”) の2つの候補が生成されている．shift した場合，各候補の末尾の翻字候補は非確定のまま残される．各文字が入力された後，各候補の翻字確率を計算し，確率の高い上位 B 個の候補のみを残す．この reduce 幅 R とビーム幅 B は，開発セットを用いた予備実験により，$R = 8, B = 32$ と固定した．

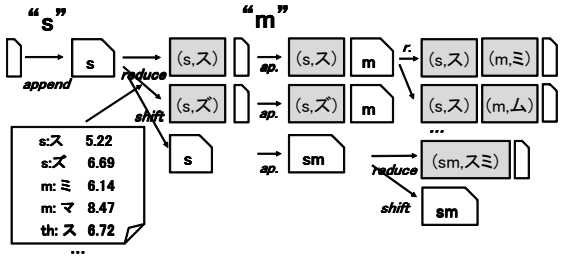

図 1: スタックデコーダにより “smith” → “スミス” を生成する過程

4 潜在意味翻字モデル

上記のアルファベータ法および JSC モデルでは，1節で述べたように，原言語の起源による違いに対処できず，学習データの全ての翻字ペアを捉えるような単一の翻字モデルを構築する．そこで，Li et al. [5] はこの問題に対して，原言語の起源や性，および姓／名の別など，翻字確率に影響を与える要因をクラス c として定義し，$\boldsymbol{s} \rightarrow \boldsymbol{t}$ の翻字確率を以下のように求めるクラス翻字モデルを提案している．

$$P_{LI}(\boldsymbol{t}|\boldsymbol{s}) = \sum_{c} P(\boldsymbol{t}, c|\boldsymbol{s}) = \sum_{c} P(c|\boldsymbol{s}) P(\boldsymbol{t}|c, \boldsymbol{s}) \qquad (3)$$

しかしながら，このクラス翻字モデルを用いるには，クラスが明示的にタグ付された学習コーパスが必要である．そこで，各翻字ペアに対して明示的なクラス c を対応づけるのではなく，潜在的なクラスを表す確率変数 z を導入し，条件付き翻字単位確率 $P(u_i|z)$ を考え，翻字確率を以下のように求める **潜在クラス翻字モデル** (LST) を定義する [2][4]：

$$P_{\mathrm{LST}}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle) = \sum_{z=1}^{K} P(z) \prod_{i=1}^{f} P(u_i|z) \qquad (4)$$

ここで，K は潜在クラス数を表す．潜在クラス z は，上記の言語起源や性別など，同じ翻字書き換え傾向を持つ翻字ペアのクラスに対応すると考えることができる．z は学習データからは直接観察されないが，以下のように EM アルゴリズムによって，学習データの尤度を最大化することにより繰り返し的に求めることができる．ここで，$\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle$ は n 番目の学習用翻字ペアである．

[1] 例えば，文字列 “abc” と “xy” の文字間アラインメントは，(a-x b-y c-ε) (a-x b-ε c-y) (a-ε b-x c-y) の 3 種類があり，例えば最初のアラインメント (a-x b-y c-ε) から，隣接する最大2つの文字アラインメントの併合を考慮すると，a-x, b-y, c-ε, ab-xy bc-y の6つの文字列間アラインメントが得られる．

[2] 例えば「英語において，有声子音に続く “ed” の音は/d/ (ド) であるが，無声子音に続くと/t/(ト) となる」ような関係．

[3] なお，評価実験では，Kneser-Ney スムージングを用い翻字単位確率をスムージングした．本稿の全てのモデルにおいて，予備実験の結果，EM アルゴリズムの繰り返しの回数は 15 回に固定した．

[4] なおここでは，$P(\boldsymbol{t}|\boldsymbol{s})$ ではなく JSC モデルのように翻字ペアの生成確率 $P(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ に潜在変数を導入したモデルになっていることに注意が必要である．この両者の性能に本質的な差は無いことを実験により確認している [2]．

・E ステップ:

$$\gamma_{nk} = \frac{\pi_k P(\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle | z=k)}{\sum_{k'=1}^{K} \pi_{k'} P(\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle | z=k')} \quad (5)$$

$$P(\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle | z) = \max_{u_1 .. u_f} \prod_{i=1}^{f_n} P(u_i|z)$$

・M ステップ:

$$\pi_k^{new} \propto \sum_{n=1}^{N} \gamma_{nk}, \quad P(u_i|z=k)^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \frac{f_n(u_i)}{f_n}$$

ここで, $N_k = \sum_n \gamma_{nk}$ であり, f_n および $f_n(u_i)$ は, それぞれ学習データ中 n 番目の翻字ペア中の翻字単位の数および翻字単位 u_i が出現する回数を表す. ここで, JSC モデルと同様, M ステップの前に, 現在の翻字モデルを用いて学習コーパスのアラインメントを更新するため, f_n は一般的に繰り返し毎に異なった値となる. さらに, 各 z に対して, $P(u_i|z)$ に従いアラインメントを更新するため, M 個の異なるアラインメントを各翻字ペアに対して保持することになり, f_n の値は実際には m に依存した f_n^m となる. なお, EM アルゴリズムの初期値としては, $P(z=k) = 1/K$ および $P(u_i|z) = P_{AB}(u) + \varepsilon$, すなわちアルファベータ法により求めた翻字単位 u の確率にランダムノイズ ε を乗せたものを用いた.

ここで, 翻字単位を単語, 翻字ペアを単語の列から構成される文書とみなすと, この LST モデルは翻字単位の多項分布の混合をもって翻字確率としており, これは文書のトピックモデルの一種であるユニグラム混合 [7] の変種として定式化が可能である. この場合, 文書(=翻字ペア)は, まずクラスを $P(z)$ に従って選択し, その後, 単語(=翻字単位)を多項分布 $P(u_i|z)$ に従って生成するという生成モデルとして捉えられる. ただし, 1節に述べたように, 本モデルは多項分布の最尤推定に基づいてパラメータを推定しているため, 学習データ中のノイズに弱く過学習しやすいという問題がある.

5 混合ディリクレ分布に基づく潜在意味翻字モデル

本節では, 前節のユニグラム混合に基づく潜在意味翻字モデル (LST) を拡張した, 混合ディリクレ分布に基づく潜在意味翻字モデル (DM-LST) を提案する. 本モデルでは, 単語の出現分布に事前分布として混合ディリクレ分布を導入することにより, 複数の潜在クラスの混合により翻字生成をモデル化するという目的を達成したまま, 最尤推定の特徴である極端な多項分布に偏るという傾向を軽減することができる. 多項分布のパラメータが混合ディリクレ分布に従う場合の合成分布は, 混合 Polya 分布:

$$\begin{aligned} P_{DM}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle) &= \int P_{Mul}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle; \boldsymbol{p}) P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^K) d\boldsymbol{p} \\ &\propto \sum_{k=1}^{K} \lambda_k P_{Polya}(\langle \boldsymbol{s}, \boldsymbol{t} \rangle; \boldsymbol{\alpha}_1^K) \quad (6) \\ &= \sum_{k=1}^{K} \lambda_k \frac{\Gamma(\alpha_k)}{\Gamma(\alpha_k + f)} \prod_{i=1}^{f} \frac{\Gamma(f(u_i) + \alpha_{ku_i})}{\Gamma(\alpha_{ku_i})} \end{aligned}$$

により与えられる [8]. ここで, P_{Mul} は $\boldsymbol{p}$ をパラメータとした多項分布, P_{DM} は混合ディリクレ分布であり, パラメータ $\boldsymbol{\alpha}_1^K = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_K)$ に従う K 個のディリクレ分布を混合比 $\lambda_1, ..., \lambda_K$ によって混合した分布である. また, $\alpha_k = \sum_u \alpha_{ku}$ である.

この時, モデルパラメータは, 以下の EM アルゴリズムにより繰り返し的に推定できる[5]:

・E ステップ:

$$\eta_{nk} = \frac{\lambda_k P_{Polya}(\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle; \boldsymbol{\alpha}_k)}{\sum_{k'} \lambda_{k'} P_{Polya}(\langle \boldsymbol{s}_n, \boldsymbol{t}_n \rangle; \boldsymbol{\alpha}_{k'})} \quad (7)$$

・M ステップ:

$$\lambda_k^{new} \propto \sum_{n=1}^{N} \eta_{nk} \quad (8)$$

$$\alpha_{ku}^{new} = \alpha_{ku} \frac{\sum_n \eta_{nk} \{f_n(u)/(f_n(u) - 1 + \alpha_{ku})\}}{\sum_n \eta_{nk} \{f_n/(f_n - 1 + \alpha_k)\}} \quad (9)$$

なお, 入力として単一の翻字ペア u が与えられた時の生成確率である予測分布 P_{DM} は, $P_{DM}(u) = \sum_{k=1}^{K} \lambda_k \alpha_{ku} / \alpha_k$ に従う. したがって, JSC モデルの更新アルゴリズムと同様, M ステップの前に, 各 k に対し, α_{ku}/α_k に従い, 学習コーパスのアラインメントを更新する. EM アルゴリズムの初期値としては, 前節同様, $\lambda_k = 1/K$, $\alpha_{ku} = P_{AB}(u) + \varepsilon$ を用いた.

LST, DM-LST 共に, 翻字生成モデルではないため, ベースラインとして3節で説明した JSC モデルおよびスタックデコーダーを用い, 翻字候補 T を生成した後, LST もしくは DM-LST を用いて候補を再ランキングすることにより, 翻字先のランク付きリストを生成した. なお, EM アルゴリズムにより学習されたパラメータは初期値によって異なるため, 同じ学習データを用いて異なる初期値から 10 個のモデル $P_{DM}^1, ..., P_{DM}^{10}$ を学習し, その平均 $\frac{1}{10} \sum_{j=1}^{10} P_{DM}^j(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ を用いて候補をランキングした.

なお, もう一つのトピックモデルとして Latent Dirichlet Allocation (LDA) があるが, LDA では文書内の各単語について異なるトピックから生成されるという仮定を置いている. これは, 翻字ペア内の翻字単位が異なる原言語から生成されるという状況に対応しており, 翻字モデルとしては適切な前提ではない[6]. 実際に予備実験により, LDA を用いて翻字をモデリングしても翻字性能は向上しないことが分かった.

6 評価実験

比較したモデルは, アルファベータ法 (AB), JSC モデル (JSC), 潜在クラス翻字モデル (LST), 混合ディリクレ分布 LST (DM-LST; 提案手法) である.

性能評価には, 翻字に関するワークショップである NEWS 2009 [3, 4] の英語→日本語カタカナ (En-Ja), 英語→中国語 (En-Ch), 英語→韓国語 (En-Ko) の各翻字データを用いた. それぞれのデータ規模を表 1の第一

[5]ここでは, leaving-one-out 法を用いた推定方法を混合分布に拡張した高速な推定手法を用いている [8].

[6]ただし, 異なる起源の混合したような語(例えば "naïveness")を除く.

表 1: 各言語ペアに対するモデルの性能比較

言語ペア	モデル	ACC	MFC	MRR
En-Ja	AB	0.293	0.755	0.378
学習:23,225	JSC	0.326	0.770	0.428
テスト:1,489	LST	0.345	0.768	0.437
	DM-LST	**0.349**	**0.776**	**0.444**
En-Ch	AB	0.358	0.741	0.471
学習:31,961	JSC	0.417	0.761	0.527
テスト:2,896	LST	0.430	0.764	0.532
	DM-LST	**0.445**	**0.770**	**0.546**
En-Ko	AB	0.145	0.537	0.211
学習:4,785	JSC	0.151	0.543	0.221
テスト:989	LST	0.079	0.483	0.167
	DM-LST	**0.174**	**0.556**	**0.237**

カラムに示した. 翻字性能の評価指標としては, 以下の3つを用いた. なお, 評価データは, 翻字元 $\boldsymbol{s}_n$ に対して, 正解として許容できる翻字先の集合である正解セット $\boldsymbol{r}_n$ が対応している. 翻字モデルによって出力された候補を, 確率の高い順に $c_{n,i}, c_{n,2}, ...$ とする.

- **ACC**(平均精度): 翻字候補トップ $1c_{n,1}$ が正解セットに含まれていれば $a_n = 1$, そうでなければ $a_n = 0$ とし, $ACC = \frac{1}{N}\sum_{i=1}^{N} s_n$ により計算される.
- **MFC**(平均 F 値): 翻字候補トップ $1c_{n,1}$ に対し, 編集距離 ED の点で最も類似した正解 $r_n^* = \arg\min_{r_{n,j}\in\boldsymbol{r}_n} ED(c_{n,1}, r_{n,j})$ との間の F 値を F_n とすると, $MFC = \frac{1}{N}\sum_{i=1}^{N} F_n$ である. ここで, F 値は, $P_n = LSC(c_{n,1}, r_n^*)/|c_{n,1}|$, $R_n = LCS(c_{n,1}, r_n^*)/|r_n^*|$, $F_n = 2R_iP_i/(R_i + P_i)$ として計算される. $|x|$ は x の文字列の長さ, $LCS(x, y)$ は x と y の最小共通部分文字列の長さであり, どちらも Unicode 文字を単位に計算される.
- **MRR**(平均順序逆数): 翻字候補リスト $c_{n,1}, c_{n,2}, \cdots$ のうち, 正解セット $\boldsymbol{r}_n$ に含まれているものの中で最も順位の高い候補を $c_{n,j}$ とすると, $MRR = \frac{1}{N}\sum_{n=1}^{N} 1/j$ である. 候補の中に正解が含まれていない場合は 0 とする.

各モデルの性能の比較を表 1に示した. ここで, LST および DM-LST の潜在クラス数 M は, 言語ペアごとに開発セットを用いて決定した. 言語ペア En-Ja, En-Ch において, 全ての評価指標で AB < JSC < LST < DM-LST であり, 提案手法の優位性を示している. また, 言語ペア En-Ko については, LST によるリランキングによって性能が JSC よりも大幅に低下しているが, これは当該言語ペアの学習データが少ないため, 過学習の影響を受けやすいためであると考えられる. また, 全ての言語ペアに対して, DM-LST において性能が最大となる M の値は, LST におけるその M の値と同じかそれよりも小さくなることが分かった. このことは, 一般的に混合ディリクレ分布はより小さい次元数で同等以上の言語モデル性能を上げるという性質 [8] とも合致している.

テストセット En-Ja 提案手法により翻字性能が改善した例には "dijon ／ディジョン" や "goldenberg ／ゴールデンバーグ" などがある. 従来手法では, それぞれ "ディヨン", "ゴールデンベルグ" が最も確率が高く, "j ／イ" "berg ／ベルグ" などの非英語的発音にモデルが影響されていることが分かり, LST についても同様の傾向がある. 提案手法では, 学習コーパスにおいて一般的な翻字傾向を保持しつつも, 複数の言語起源に対応できていることが分かる. この傾向はテストセット En-Ch 中の "covell ／科夫尔 (kefuer) →科维尔 (keweier)" "netherwood ／内特赫伍德 (neitehewude) →内瑟伍德 (neisewude)", における英語発音 "ve ／维 (wei)" "th ／瑟 (se)" や, En-Ko 中の darling ／다르링 (dareuling) → 달링 (dalling) などにも見られた.

一方で, En-Ch 中の "gutheim ／古特海姆 (gutehaimu), En-Ko 中の martina ／마르티나 (mareutina) などの非英語的発音をもつ語についても正しく翻字ができており, 語の起源により適切なモデルを適用できていることが分かる. ただ, これら非英語的発音を持つ語の翻字は一般には文脈依存であり(例えば, "charles" には "チャールズ"(英) と "シャルル"(仏) の読みがある), 精度をより高めるには文脈の考慮や Web 統計の利用などが必要であろう.

7 おわりに

本研究では, 原言語の起源をモデル化した潜在クラス翻字法を拡張したディリクレ分布に基づく潜在クラス翻字モデルを提案した. 評価実験の結果, 過学習を抑えながら, 全体の翻字性能を向上させることができることが分かった. 翻字単位のバイグラム以上の依存関係をどのようにして潜在クラスを用いてモデル化するかは今後の課題である.

参考文献

[1] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling. In *Proc. ACL-2000*, pp. 286–293, 2000.

[2] Masato Hagiwara and Satoshi Sekine. Latent class transliteration based on source language origin. In *Proc. of ACL-HLT 2011*, pp. 53–57, 2011.

[3] Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. Report of news 2009 machine transliteration shared task. In *Proc. of the 2009 Named Entities Workshop*, pp. 1–18, 2009.

[4] Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine. Whitepaper of news 2009 machine transliteration shared task. In *Proc. of the 2009 Named Entities Workshop*, pp. 19–26, 2009.

[5] Haizhou Li, Zhang Min, and Su Jian. A joint source-channel model for machine transliteration. In *Proc. of ACL 2004*, pp. 159–166, 2004.

[6] Haizhou Li, Khe Chai Sum, Jin-Shea Kuo, and Minghui Dong. Semantic transliteration of personal names. In *Proc. of ACL 2007*, pp. 120–127, 2007.

[7] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, Vol. 39, No. 2, pp. 103–134, 2000.

[8] 山本幹雄, 貞光九月, 三品拓也. 混合ディレクレ分布を用いた文脈のモデル化と言語モデルへの応用. 情報処理学会研究報告. SLP, 音声言語情報処理, Vol. 2003, No. 104, pp. 29–34, 2003.