文字の文脈を考慮したカタカナから英語への Back Transliteration

後藤 功雄, 加藤 直人, 江原 暉将
NHK 放送技術研究所
{goto.i-es, katou.n-ga, ehara.t-eo}@nhk.or.jp

1 はじめに

英語と日本語など文字が異なる言語間では, 固有名詞は多くの場合, 元の単語の発音を表す外来語に翻字により翻訳される. 翻字された単語は, 辞書に登録がない場合も多く, 元の言語に翻訳することは困難である. そこで, 辞書に登録がない場合でも, 翻字された単語を自動的に元の言語へ戻す翻字(Back transliteration)ができれば, 言語横断情報検索などに有効である.

本稿では, カタカナに翻字された単語を元の英単語に Back transliteration する手法について述べる. 本手法は, 部分文字列単位で英単語を生成することで, 学習データに存在しない英単語も生成することが可能である. また, 本手法は, これまでに提案されている手法に比べ, 英語とカタカナの対応確率を求める際に, 確率の推定に有効な英語の文脈情報を用いて, 精度の向上を行っている. さらに, 英単語の生成確率を求める際に, 1 文字単位の N-gram によって英語の文脈情報を用いる. 文脈情報を利用する際には, 最大エントロピー法に基づいた確率モデルを用いる. これによって, 文脈情報を有効に利用することができる.

Back transliteration は 1.1 式により行うことができる.

$$\arg\max_{English} P(English \mid Japanese\ katakana) \tag{1.1}$$

1.1 式をベイズの定理により変形すると, 1.2 式が得られる.

$$= \arg\max_{English} P(English)P(Japanese\ \ katakana \mid English) \tag{1.2}$$

ベイズの定理により, 式を変形する理由は, 英語の綴りを条件として発音を示すカタカナを推定する翻訳モデル $P(Japanese\ \ katakana \mid English)$ を導入するためと, 言語モデル $P(English)$ と翻訳モデルを分離して導入するためである. 翻字によって生成されたカタカナ表現は元の単語の発音を表現していることから, 1.2 式の翻訳モデルは, 英語の発音を推測するモデルと同じと考えることができる. そこで, Back transliteration を行う際に重要な問題は, 言語モデルによる英語らしさを評価する部分と, 翻訳モデルによる英単語の発音を推定する部分となる. 英語の単語の各部分の発音の推定は難しいため, 文脈情報を利用して精度を上げることが重要である.

Back transliteration の手法は, 過去にいくつかの方法が提案されている. Knight ら(1998)は, 日本語から英語への変換を提案している. しかし, この手法は, 最も困難な英語の発音の推定には, 英語の発音辞書の見出しをそのまま利用している. 文字単位の処理を行っているのは, 英語の発音と日本語の発音の対応確率を求める部分のみである. そのため, 英語の発音辞書に登録されていない単語を扱うことができない.

発音辞書を用いない手法もある. Fujii ら(2001)は, 言語横断検索で検索質問を翻訳する際に Back transliteration を用いて日本語を英語に変換している. 彼らの手法は, 生成する英単語の候補があらかじめ得られていることを仮定している.

事前に得られていない単語を生成できる手法もある. Jeong ら(1999)は, 韓国語から英語に Back transliteration する手法を提案している. ここで, 翻訳元言語の単語 S を 1.3 式, 翻訳先言語の単語 T を 1.4 式のように表現する.

$$S = s_1 s_2 ... s_l = su_1 su_2 .. su_n \tag{1.3}$$

$$T = t_1 t_2 ... t_m = tu_1 tu_2 .. tu_n \tag{1.4}$$

s_j は翻訳元言語の j 番目の文字, t_j は翻訳先言語の j 番目の文字である. また, su_i は翻訳元言語の i 番目の部分文字列, tu_i は su_i に対応する翻訳先言語の i 番目の部分文字列である. 彼らの手法は, 1.5 式による.

$$\arg\max_{T} \prod_{i} P(tu_i \mid tu_{i-1}) P(su_i \mid tu_i) \tag{1.5}$$

言語モデル $P(tu_i \mid tu_{i-1})$ で部分文字列 tu を単位として扱うことにより, 事前には得られない英語の単語も生成することができる. しかし, 言語モデルで翻訳先言語の文脈を考慮する際に部分文字列の 2-gram を用いているので, 部分文字列の長さを大きくすると, モデルが対応できるデータが過疎になってしまう. それに対して, 我々は, 1 文字単位の N-gram を用いている. また, Jeong らの翻訳モデル $P(su_i \mid tu_i)$ では, 文脈情報を考慮していない.

決定木を用いて翻訳元の発音を示す文脈を考慮して韓国語から英語への Back transliteration を行う手法が提案されている(Kang et al., 2000). この手法は, 1.6 式により韓国語の文字の構成要素単位で変換を行う.

$$\begin{aligned} &\arg\max_{tpu_1} P(tpu_1 \mid s_{-3}, s_{-2}, s_{-1}, s_1, s_2, s_3, s_4)... \\ &\arg\max_{tpu_l} P(tpu_l \mid s_{l-3}, s_{l-2}, s_{l-1}, s_l, s_{l+1}, s_{l+2}, s_{l+3}) \end{aligned} \tag{1.6}$$

ここでは, s_i は韓国語の文字の構成要素を示し, tpu_i は, s_i に対応する英語の部分文字列を示す. この手法では, 英語の部分文字列を推定する際に, 発音を示す韓国語の文脈情報を用いている. しかし, 発音の文脈情報は未知の英単語の推定に対してあまり有効な情報にはならない. 例えば, 発音を示す表現 KURO-DO の O-に対応する英語の部分文字列を推定する場合を考える. この時に, 前後の発音の情報 KURO-DO も得たとする. しかし, この情報だけでは, O-に対応する英語の部分文字列が, "a", "o", "au", "aw", "ho", "oa", "oe", "oh", "or", "ou", "ow", "oer"など多数考えられる候補のうちのどれになるかを判別することができない. よって, 発音の文脈情報は, 英語の部分文字列を推定する際の有効性は低い. それに対して, 我々は, 英語の発音を推定する際に, 発音の推定に有効な英語の文字列の文脈情報を用いている.

2 提案する手法

本手法による翻字処理を実行フェーズと学習フェーズに分けて述べる.

2.1 実行フェーズ

2.1.1 変換候補のラティス作成

カタカナの単語から英語の単語を直接推定する. Back transliteration を行うカタカナの単語の先頭に^, 単語の末尾に$を追加して, カタカナの単語 K を 2.1 式のように表現する.

$$K = k_0 k_1 ... k_{m+1} \tag{2.1}$$

$$k_0 = \text{^}, k_{m+1} = \$ \tag{2.2}$$

ここで, k_j はカタカナの単語の j 番目の文字であり, m は, カタカナの単語の^と$以外の文字数である. このカタカナの単語の各部分に対して, 対応付けされたカタカナの部分文字列 ku (katakana unit)と英語の部分文字列 eu (English unit)からなる変換候補生成規則の集合を用いて, カタカナの単語内の各部分に対応する英語の部分文字列を得る.

この際に, 変換ルールには複数の長さの部分文字列が含まれるため, 例えば, チェイニー→チェ/イ/ニー, チェイ/ニーのよ

うに，変換ルールを適用する区切りに曖昧性が生ずる．これらの曖昧性を持つ変換候補からラティス $\mathcal{L}\{E\}$ を作成することで，曖昧性を持つ変換候補を統一的に扱うことができる．変換候補生成規則の適用方法は，K の文字列中に一致する変換候補生成規則の ku を全て適用し，その ku に対応する全ての eu により，$\mathcal{L}\{E\}$ を作成する．図1にチェイニーのラティス $\mathcal{L}\{E\}$ の例を示す．

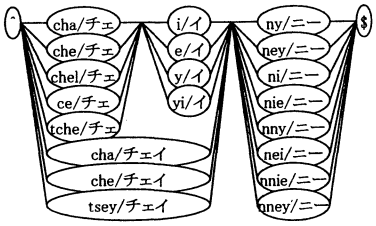

図1 変換候補のラティス $\mathcal{L}\{E\}$ の例

$\mathcal{L}\{E\}$ 中の^から$までの各経路 $p_d \in (p_1, p_2, \ldots p_q)$ 中の部分文字列をつないだ文字列が英単語の候補となる．ここで，q は，$\mathcal{L}\{E\}$ 中の^から$までの経路数を示す．

ここで，$\mathcal{L}\{E\}$ 中のある経路 p_d を選択した場合を考える．この場合の p_d 中の^と$以外の部分文字列の数を $n(p_d)$ とする．また，p_d 中の部分文字列に，先頭から順番に番号を付与する．すると，p_d に対する，カタカナの単語 K とその変換結果の英語の単語 E は，次のようになる．

$$K = k_0 k_1 .. k_{m+1} = ku_0 ku_0 .. ku_{n(p_d)+1} \tag{2.3}$$

$$E = e_0 e_1 ... e_{l(p_d)+1} = eu_0 eu_1 .. eu_{n(p_d)+1} \tag{2.4}$$

$$\begin{cases} k_0 = e_0 = ku_0 = eu_0 = \text{^} \\ k_{m+1} = e_{l(p_d)+1} = ku_{n(p_d)+1} = eu_{n(p_d)+1} = \$ \end{cases} \tag{2.5}$$

ここで，e_j は英語の単語の j 番目の文字であり，$l(p_d)$ は英語の単語の^と$以外の文字数である．$\mathcal{L}\{E\}$ 中の各 p_d における 2.4 式の $eu_0 eu_1 .. eu_{n(p_d)+1}$ が英単語の候補となる．また，2.3 式の $ku_0 ku_1 .. ku_{n(p_d)+1}$ が 2.4 式の候補を出力する際のカタカナの部分文字列の区切れを示す．

2.1.2 英単語の候補の評価式

カタカナの単語を入力として，対応する英語の単語を推定するためには，2.6 式を満たす E を求めればよい．

$$\arg\max_E P(E \mid K) \tag{2.6}$$

ベイズの定理を用いて変形する．

$$= \arg\max_E P(E) P(K \mid E) \tag{2.7}$$

2.7 式を直接求めることは未知の単語に対して難しい．そこで，2.3 式，2.4 式により，2.7 式中の翻訳モデル $P(K \mid E)$ の単語を部分文字列に分解する．

$$= \arg\max_E P(E) P(ku_0, ku_1, .., ku_{n(p_d)+1} \mid eu_0, eu_1, .., eu_{n(p_d)+1}) \tag{2.8}$$

$$= \arg\max_E P(E) \prod_{i=1}^{n(p_d)} P(ku_i \mid ku_1, ku_2, .., ku_{i-1}, eu_0, eu_1, .., eu_{n(p_d)+1}) \tag{2.9}$$

2.9 式では，$\mathcal{L}\{E\}$ 中の全ての経路 $p_d \in (p_1, p_2, \ldots p_q)$ から探索を行う．2.9 式の翻訳モデルのカタカナの接続に関する条件を削除して近似する．

$$\approx \arg\max_E P(E) \prod_{i=1}^{n(p_d)} P(ku_i \mid eu_0, eu_1, .., eu_{n(p_d)+1}) \tag{2.10}$$

さらに，翻訳モデルの英語の文脈情報を eu_i の前 a 文字，後 b 文字のみに近似する．

$$\approx \arg\max_E P(E) \prod_{i=1}^{n(p_d)} P(ku_i \mid e_{start(i)-a}, .., e_{start(i)-2}, e_{start(i)-1}, eu_i, e_{end(i)+1}, e_{end(i)+2}, .., e_{end(i)+b}) \tag{2.11}$$

ここで，$start(i)$ は i 番目の部分文字列 eu_i の初めの文字の位置で，$end(i)$ は i 番目の部分文字列 eu_i の最後の文字の位置である．a と b は定数である．さらに，言語モデル $P(E)$ を文字の N-gram($N = c+1$)で近似する．c は定数である．

$$\approx \arg\max_E \prod_{j=1}^{l(p_d)+1} P(e_j \mid e_{j-c}, .., e_{j-2}, e_{j-1}) \times \prod_{i=1}^{n(p_d)} P(ku_i \mid e_{start(i)-a}, .., e_{start(i)-2}, e_{start(i)-1}, eu_i, e_{end(i)+1}, e_{end(i)+2}, .., e_{end(i)+b}) \tag{2.12}$$

2.12 式が，我々の手法で用いる評価式である．

2.12 式の翻訳モデル $P(ku_i \mid ..., e_{start(i)-1}, eu_i, e_{end(i)+1}, ...)$ や，言語モデル $P(e_j \mid e_{j-c}, .., e_{j-2}, e_{j-1})$ で英語の文脈情報を考慮する際に，単純に行うとモデルが対応できるデータが過疎になってしまう．そのため，これらのモデルには，最大エントロピー法に基づいて構築した確率モデルを利用する．最大エントロピー法に基づいた確率モデルでは，英語の文脈情報の文字を1つの属性として捉え，それらを組み合わせた素性を用いることにより，文脈情報を有効に利用することができる．用いた素性については，2.2.4 節で述べる．

2.2 学習フェーズ

2.2.1 単語内の部分文字列が対応付けされたコーパス作成

カタカナと英語の日英対訳の辞典から，単語内部の部分文字列の対応をつけたコーパスを半自動により作成した．対応付けの半自動化の部分の手法の概要は次のとおりである．日本語と対訳の英語の[シソーラス：thesaurus]の単語内対応付けを行う例を示す．

1) カタカナをローマ字に変換する．[shiso-rasu：thesaurus]
2) 人手により作成したローマ字の部分文字と英語の部分文字で対応する可能性の高いリスト[1]を用いて，英語とローマ字が単語全体で最適に一致する部分文字列対応を決定する[2]．その際に，単語中の位置や順番の情報も利用する．さらに残りの部分を対応づける．対応する文字列がない場合は，前の文字列にマージする．[sh/i/s/o-/r/a/s/u：th/e/s/a/u/r/u/s]
3) ローマ字をカタカナに戻す．文字の間に分割点が含まれる場合は，その分割点と対応する英語の分割点を削除する．[シ/ソー/ラ/ス：the/sau/ru/s]
4) ポストフィルターにより分割点の修正と追加を行う．最後に人手により間違いを修正してコーパスを完成させる．[シ/ソー/ラ/ス：the/sau/ru/s]

本手法による部分対応付け結果は，NULL 文字への対応を含まない．また，発音を表現するカタカナと対応させることによって英語の部分文字列の単位を決定しているので，英語の部分文字列の単位は，発音を考慮した単位となる．

---

1 153 個の対応規則と英語文字 x との対応の例外規則を用いた

2 促音を示す「ッ」に相当する文字については，後の文字列にマージし，対応付けの対象としない．

2.2.2 変換候補生成規則の作成

単語内の部分文字列の対応がついたコーパスを用いて，カタカナと英語の部分文字列を対応付けした変換候補を生成する規則を作成する．例えば，[シ/ソー/ラ/ス：the/sau/ru/s]のデータからは，"シ"→"the"，"ソー"→"sau"，"ラ"→"ru"，"ス"→"s"という変換候補生成規則を得る．学習データ中の全ての英語とカタカナの部分文字列の対応から，変換候補生成規則を作成する．

2.2.3 最大エントロピー法による確率モデルの学習

最大エントロピー法による学習は，与えられた制約を満たすモデルの中で最も一様な分布であるモデルを選択するものである．ここで分布の一様さは，確率モデルのエントロピー $H(P)$ を用いる．

$$H(P) = -\sum_{x,y} P(x,y)\log P(x,y) \tag{2.14}$$

ここで，y は $\boldsymbol{ku}_i$，x は $e_{start(i)-a},...,e_{start(i)-1},\boldsymbol{eu}_i,e_{end(i)+1},...,e_{end(i)+b}$ を示す．$P(x,y)$ は，x と y の同時確率分布を表す．また，モデルは，素性関数による制約を満たしていなければならない．x と y に関する条件 k 個を $condition(x,y)_i$，$i \in \{1,2,..,k\}$ として，素性関数を次のように定義する．

$$f_i(x,y) = \begin{cases} 1 & (x,y) \in condition(x,y)_i \\ 0 & (\text{それ以外}) \end{cases} \tag{2.15}$$

学習データ中の同時確率分布(経験的確率分布)を $\tilde{P}(x,y)$ と表現すると，$P(x,y)$ に対する制約は，$i \in \{1,2,....,k\}$ に対して

$$\sum_{x,y} P(x,y)f_i(x,y) = \sum_{x,y} \tilde{P}(x,y)f_i(x,y) \tag{2.16}$$

となる．ここで，求めたい確率モデルは，条件付確率 $P(y \mid x)$ である．(2.16)式を変形し，$P(x)$ を近似して $\tilde{P}(x)$ を用いた．

$$\sum_{x,y} P(y \mid x)\tilde{P}(x)f_i(x,y) = \sum_{x,y} \tilde{P}(x,y)f_i(x,y) \tag{2.17}$$

制約を満たすモデルの集合を $\mathcal{P}$ とすると，推定する確率モデル P^* は，$\mathcal{P}$ の中で，エントロピーを最大にするものである．

$$P^* = \arg\max_{P \in \mathcal{P}} H(P) \tag{2.18}$$

2.17 式を制約条件として，2.18 式を満たす確率モデル P^* を求めた．モデルのパラメータの推定は，Berger (1996)の方法を用いた．

2.2.4 最大エントロピー法で利用する素性

2.12 式中の翻訳モデルと言語モデルで利用する素性関数を定義する．翻訳モデルで利用する英語の文脈情報は，変換対象の部分文字列の前後 3 文字とし，$a = b = 3$ とした．翻訳モデルでは，文字情報だけでなく，子音，母音，半母音の区別の情報も利用した．e_j の子音，母音，半母音の区別の情報を $G(e_j)$ と表す．言語モデルで利用する英語の文脈情報は，e_j の前 3 文字とし，$c = 3$ とした．$\boldsymbol{eu}_i, e_j, G(e_j), \boldsymbol{ku}_i$ をそれぞれ1つの属性として，それらの属性を組み合わせることにより素性関数を定義した．距離が近いことと，連続していることが重要であると考え，この重要度を元に，属性の組み合わせを作成した．翻訳モデルの属性の組み合わせの種類を表 1 に，言語モデルの属性の組み合わせの種類を表 2 に示す．

3 実験

3.1 学習データとテストデータ

Back transliteration の処理対象をカタカナで表現された外国の人名として，実験を行った．

まず，実験に用いた学習データについて述べる．翻訳モデルの学習には，日外アソシエーツの「8万人西洋人名よみ方綴り方辞典」[3]のデータを用いた．この中から，a-z のアルファベットで表現される人名の単語とその対訳のカタカナの人名の単語を用いた．1 単語対を 1 データとして，83,086 単語対のデータを用いた．カタカナの単語の異なり数は 39,579，英語の単語の異なり数は 39,799 であった．これらの単語内の部分文字列の対応付けを行った．対応付けされた部分文字列の平均文字数は，カタカナが 1.30 文字，英語が 1.86 文字であった．言語モデルの学習には，アメリカの商務省の国勢調査局が 1990 年の国勢調査結果から作成したアメリカの人名リスト[4]を利用した．このリストには，頻度情報が付与されているので，言語モデルの学習には，頻度情報も利用して行った．言語モデルの学習に用いた人名リストの単語の異なり数は 91,910 であった．

次にテストデータについて述べる．テストデータには，2002 年の共同通信社の世界年鑑に記載されていたアメリカの閣僚と政府・官僚の人名の 156 単語とカナダ，イギリス，オーストラリア，ニュージーランドの閣僚の人名の 177 単語の合計 333 のカタカナの単語を用いた．それらの人名の対訳の英単語を正解として用いた．英単語に a-z 以外の文字を含む単語は除外している．姓と名の区別は付けずに，1 単語を 1 データとした．テストデータのカタカナの単語の平均文字数は 4.2 文字，正解の英単語の平均文字数は 6.0 文字である．これらのテストデータのうち，翻訳モデルの学習データに存在しないカタカナの単語をデータ1，言語モデルの学習データに正解の英訳語が存在しないカタカナの単語をデータ2とした．データ1の単語数は 58 単語，データ2の単語数は 28 単語であった．

表 1 翻訳モデルの素性関数を定義する条件の種類

	属性の種類							
条件の種類	y	x						
属性の組み合わせの種類	ku_i				eu_i			
	ku_i				eu_i	$e_{end,j+1}$		
	ku_i				eu_i	$\alpha e_{end,j+1}$		
	ku_i				eu_i	$e_{end,j+1}$	$e_{end,j+2}$	
	ku_i			$e_{start,j-1}$	eu_i	$\alpha e_{end,j+1}$	$\alpha e_{end,j+2}$	
	ku_i			$\alpha e_{start,j-1}$	eu_i	$e_{end,j+1}$	$e_{end,j+2}$	$e_{start,j+3}$
	ku_i		$e_{start,j-2}$	$e_{start,j-1}$	eu_i	$\alpha e_{end,j+1}$	$\alpha e_{end,j+2}$	$\alpha e_{end,j+3}$
	ku_i		$\alpha e_{start,j-2}$	$\alpha e_{start,j-1}$	eu_i			
	ku_i	$e_{start,j-3}$	$e_{start,j-2}$	$e_{start,j-1}$	eu_i			
	ku_i	$\alpha e_{start,j-3}$	$\alpha e_{start,j-2}$	$\alpha e_{start,j-1}$	eu_i			
	ku_i			$e_{start,j-1}$	eu_i	$e_{end,j+1}$		
	ku_i			$\alpha e_{start,j-1}$	eu_i	$\alpha e_{end,j+1}$		
	ku_i			$e_{start,j-1}$	eu_i	$e_{end,j+1}$	$e_{end,j+2}$	
	ku_i			$\alpha e_{start,j-1}$	eu_i	$\alpha e_{end,j+1}$	$\alpha e_{end,j+2}$	
	ku_i		$e_{start,j-2}$	$e_{start,j-1}$	eu_i	$e_{end,j+1}$		
	ku_i		$\alpha e_{start,j-2}$	$\alpha e_{start,j-1}$	eu_i	$\alpha e_{end,j+1}$		

表 2 言語モデルの素性関数を定義する条件の種類

	属性の種類			
条件の種類	y	x		
属性の組み合わせの種類	e_j	e_{j-3}	e_{j-2}	e_{j-1}
	e_j		e_{j-2}	e_{j-1}
	e_j			e_{j-1}

3.2 確率モデルの学習

最大エントロピー法で用いる素性関数は，翻訳モデルでは，表 1 に示した属性の組み合わせのうち 1 回以上観測された属性の組み合わせから素性関数を作成した．素性関数の数は 473,407 個である．言語モデルでは，表 2 に示した属性の組み合わせのうち，スムージングを考慮して学習データに 3 回以上観測された属性の組み合わせから素性関数を作成した．素性関数の数は 38,679 個である．Berger(1996)の方法によるモデルのパラメータ推定の繰り返し回数は 500 回行った．

---

[3] 1994 年に出版されたもの

[4] http://www.census.gov/genealogy/names/ から入手することができる

3.3 比較のための手法

比較のために以下に示す手法による実験も行った.

- 手法 A

翻訳モデルの学習に用いた対訳の学習データにより, テストデータを英語に変換する. 同じカタカナの単語に対して複数の訳語が存在する場合は, ランダムに訳語を選択する.

- 手法 B

英語の文脈情報を用いない 3.1 式による手法.

$$\arg\max_{E} \prod_{i=1}^{n(p_d)} P(\boldsymbol{eu}_i \mid \boldsymbol{ku}_i) \tag{3.1}$$

- 手法 C

英語の文脈情報は用いず, 発音を示すカタカナの文脈情報を考慮する 3.2 式による手法. 文脈情報の利用は, 本手法と同じ方法[5]による.

$$\arg\max_{E} \prod_{i=1}^{n(p_d)} P(\boldsymbol{eu}_i \mid k_{start(i)-3},.,k_{start(i)-1},\boldsymbol{ku}_i,k_{end(i)+1},.,k_{end(i)+3}) \tag{3.2}$$

- 手法 D

本手法の言語モデルを用い, 翻訳モデルは文脈情報を考慮しない $a=b=0$ とした 3.3 式による手法.

$$\arg\max_{E} \prod_{j=1}^{l(p_d)+1} P(e_j \mid e_{j-3},e_{j-2},e_{j-1}) \prod_{i=1}^{n(p_d)} P(\boldsymbol{ku}_i \mid \boldsymbol{eu}_i) \tag{3.3}$$

3.4 結果

本手法による Back transliteration の結果[6]と **3.3** 節で示した手法による結果を表 3, 表 4, 表 5 に示す. 表 3 は, 全テストデータを対象にした結果, 表 4 はデータ1を対象にした結果, 表 5 はデータ2を対象にした結果である. 生成した英単語が正解の英単語に完全に一致した場合に成功とした.

表 3 全テストデータの結果

	上位解が正解を含む率(%)					
	1位	2位以上	3位以上	5位以上	10位以上	20位以上
手法A	63.7	73.3	75.4	76.3	76.6	76.6
手法B	23.7	34.5	42.6	53.8	63.4	71.5
手法C	40.2	53.2	59.8	66.7	75.1	79.0
手法D	57.1	70.3	73.0	76.6	81.7	86.2
本手法	62.2	71.8	78.7	82.6	87.4	90.1

表 4 データ1の結果

	上位解が正解を含む率(%)					
	1位	2位以上	3位以上	5位以上	10位以上	20位以上
手法A	0.0	0.0	0.0	0.0	0.0	0.0
手法B	6.9	17.2	24.1	31.0	43.1	46.6
手法C	8.6	13.8	17.2	22.4	32.8	36.2
手法D	17.2	29.3	32.8	36.2	53.4	60.3
本手法	17.2	31.0	39.7	51.7	60.3	65.5

表 5 データ2の結果

	上位解が正解を含む率(%)					
	1位	2位以上	3位以上	5位以上	10位以上	20位以上
手法A	17.9	21.4	25.0	25.0	25.0	25.0
手法B	10.7	14.3	25.0	32.1	42.9	50.0
手法C	14.3	21.4	25.0	28.6	39.3	46.4
手法D	7.1	14.3	14.3	14.3	25.0	46.4
本手法	10.7	17.9	25.0	32.1	60.7	67.9

提案手法による全テストデータの 1 位の結果の正解率は 62.2%であり, 上位 20 位の正解率は 90.1%であった. ここでは, 正解が 1 位でない場合でも, 上位の単語が必ずしも存在しないものではなく, 多くの場合, 存在する人名である可能性が高いと思われる. 全テストデータの上位 10 位の単語正解率は, 手法 A に比べ 11%, 手法 B に比べ 24%, 手法 C に比べ 12%, 手法 D に比べ 6%程度の向上が見られ, 本手法の英語の文脈を考慮した Back transliteration の手法の有効性が確認された.

本手法の言語モデルを利用している手法 D と, 英語の文脈情報を用いない手法 B を比較すると, 手法 D の精度が良いことから, 本手法で用いた文字の 4-gram という長い文脈を考慮する言語モデルが有効であることが分かる. さらに, 本手法と手法 D との比較から, 英語の文脈情報を考慮する翻訳モデルが有効であることが分かる.

データ1の結果より, 本手法は学習データに存在しないカタカナの単語に対応できることが分かる. また, データ1の結果において, 手法 B と比較して手法 C の精度が低下していることから, 発音を示す文脈情報は, 未知の単語に対して有効性が低いことが分かる.

データ2の結果より, 本手法は学習データに存在しない英語の単語もある程度上位の結果として生成できることが分かる.

4 おわりに

本稿では, 統計的手法を用いてカタカナの単語から英語の単語を生成する Back Transliteration の手法について述べた. 本手法により, 学習データに存在しない英語の単語も生成することが可能となる. 本手法は, これまでに提案されている手法に比べ, 英語とカタカナの対応確率を求める際に, 対応確率の推定に有効な英語の文脈情報を利用して, 精度の向上を行っている. また, 英語の単語の生成確率を文字の 4-gram という長い文脈情報を利用して求めている. 文脈情報の利用には, 最大エントロピー法に基づいた確率モデルを用いた. これによって文脈情報を有効に利用することができる. 英語圏の人名を対象にした実験で, 上位 10 位の単語正解率は 87%であった. これは, 対訳の学習データにより直接変換した場合(手法 A)と比較して 11%, 英語の文脈情報を利用しない手法(手法 B)と比較して 24%程度高い精度であり, 本手法の有効性が確認された.

5 ただし, 子音・母音・半母音の区別情報は用いない.

6 2.12 式で $\mathcal{L}\{E\}$ 中の全ての経路 $p_d \in (p_1, p_2, ... p_q)$ から探索を行うことは, 経路数が多い場合に現実的ではない. そのため, 2.12 式で, $a=b=0$, $c=3$ とした評価式による上位 300 位の経路と, $\mathcal{L}\{E\}$ 中の経路が示す単語が言語モデルの学習データに存在する場合の経路から探索を行った. $b=0$ の場合は, 動的計画法に基づいて効率的に上位解を求めることができる.

参考文献

Kevin Knight and Jonathan Graehl. 1998. *Machine Transliteration.* Computational Linguistics, Vol.24, No.4, pp.599-612.

Atsushi Fujii and Tetsuya Ishikawa. 2001. *Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration.* Computers and the Humanities, Vol.35, No.4, pp.389-420.

Kil Soon Jeong, Sung Hyun Myaeng, Jae Sung Lee and Key-Sun Choi. 1999. *Automatic Identification and Back-Transliteration of Foreignn Words for Information Retrieval.* Information Processing and Management, Vol.35, No.4, pp.523-540.

Byung-Ju Kang and Key-Sun Choi. 2000. *Automatic Transliteration and Back-Transliteration by Decision Tree Learning.* International Conference on Language Resources and Evaluation. pp.1135-1411.

Adam L. Berger, Stephen A. Della Pietra, Vincent J. Della Pietra. 1996. A *Maximum Entropy Approach to Natural Language Processing.* Association for Computational Linguistics, Vol.22, No.1, pp.39-71.