

英単語のアルファベット表記から仮名表記への変換

堀内 雄一 山崎 一生
(新潟大学工学部)

英単語を日本語として借用する場合、英単語の発音のまま借用し、それを片仮名で表記する
場合が多い。ここでは、英語と日本語の音韻体系について考察し、英単語の綴り字から置き換
え規則を用いて片仮名へ自動変換する方法について述べる。この方法は単音節語の変換につい
ては有効であることがわかった。

A METHOD FOR TRANSLITERATING
THE SPELLING OF ENGLISH WORDS INTO "KATAKANA"
USING THE REWRITE RULES

Horiuchi Yūichi Yamasaki Issei
Faculty of Engineering, Niigata University
Ikarashi 2-nochō, Niigata 950-21, Japan

Abstract

When we use English words as Japanese, in most cases, we borrow the pronunciations of the English words and write those in "katakana". This paper considers the phonological system of English and that of Japanese, and describes a method which transliterates automatically the spelling of English words into "katakana" using the rewrite rules. Proposed method can be used to the transliterations of monosyllables.

1 はじめに

外国語を音のまま借用しようとする場合、その外国語の発音を自国語の音韻体系（発音体系）にあわせて発音を変えて借用しているものと考えられる。例えば、英語の <romantic> /roumántik/ は、中国語に借用されると <罗曼蒂克> /luómándìkè/ となり、日本語に借用されると <ロマンチック> /romanchikku/ あるいは <ロマンティック> /romantikku/ という具合になる⁽¹⁾。このように外国語を音のまま借用する場合、両言語の音韻体系間に規則性があるものと考えられる。例えば、英語の子音 /r/ は、中国語では /l/ に対応し、日本語では /r/ に対応するというような関係がある。

ここでは、この借用語における二言語の音韻体系間の規則性に注目し、置き換え規則を用いて英単語のアルファベット表記（原綴り）から片仮名表記へ自動変換する方法について述べる。

2 英語と日本語の音韻体系

2-1 音素

英語と日本語における音韻体系の構成要素である音素を表1に示す⁽²⁾⁽³⁾⁽⁴⁾。

表1(a) 英語の音素

子音音素	母音音素
p, b; t, d; k, g	i, e, æ, a, o, u, ə
f, v; θ, ð; s, z; š, ž; h	
č, j	
m, n, ŋ	
l	
r, y, w	

表1(b) 日本語の音素

子音音素	母音音素
p, b; t, d; k, g	i, e, a, o, u
s, z; h	
m, n	
r	
y, w	
Q, N (モーラ音素)	

この図を見てすぐにわかるように、英語の音素数が日本語の音素数より多い。このことから英語と日本語の音素の対応関係において、当然日本語の一つの音素が英語の複数の音素に対応するものと考え

英語音素		日本語音素						
p	→	p	š	→	s i	i	→	i
b	→	b	ž	→	z i	e	→	e
t	→	t	h	→	h	æ	→	a
d	→	d	č	→	t i	o	→	o
k	→	k	j	→	z i	u	→	u
g	→	g	m	→	m	a	→	a
f	→	h u	n	→	n	ə	→	a
v	→	b	ŋ	→	n g u			
θ	→	s	l	→	r			
ð	→	z	r	→	r			
s	→	s	y	→	y			
z	→	z	w	→	w			

図1 英語の音素の日本語の音素との対応関係

えられる。例えば、英語の二つの子音音素 /r/ と /l/ は、日本語の子音音素 /r/ に対応する。英語の音素の日本語の音素への対応の概略を図1に示す。

2-2 音節構造

どの言語にも見られる音節構造は、子音+母音（子音を C、母音を V と表わすと CV）である。この音節構造を開音節と呼ぶ。日本語の音節構造はほとんどこの開音節からなっている（表2参照）。一方、英語の音節構造は、CV のほかに、表2に示すように、母音を中心に音節の始めには、最

表2 英語と日本語の音節構造

日本語	英語			
CV	V	CV	CCV	CCCV
CSV	VC	CVC	CCVC	CCVC
CVV	VCC	CVCC	CCVCC	CCCVCC
CSVV	VCCC	CVCCC	CCVCCC	
CVM	VCCCC	CVCCCC	CCVCCCC	
CSVM				

C：子音音素，S：半母音 /j, w/, V：母音音素，M：モーラ音素 /ɑ, ɱ/

高3つまでの子音が連続することができ、音節の終わりには、最高4つまでの子音が連続することができる¹⁾。このように英語の音節構造は、日本語の音節構造と比較して非常に複雑であることがわかる。日本語では、母音を後にもたない子音は、促音とはつ音を除いて存在しない。そのため英語の発音における語末の子音や子音群には、図2に示すように母音が添えられることになる。図2(1)

英語：	C V C	C C V C C C
	↓ ↓ ↓	↓ ↓ ↓ ↓ ↓ ↓
日本語：	C V CV	CV C V CV CV CV
	(1)	(2)

図2 母音が添えられる音節構造の例

は語末の子音に母音が添えられる音節構造の例で、図2(2)は子音群に母音が添えられる音節構造の例である。この添えられる母音は、子音の種類によって決定される。表3に各子音の後に添えられる母音を示す。英語の子音 /p, b, k, g, f, v, θ, ð, s, z, ʃ, ʒ, m, ŋ, l/ には /u/ が、/t, d/ には /o/ が、/ç, j/ には /i/ が添えられる。

表3 子音に添えられる母音

添えられる母音	英語の子音
u	p, b, k, g, f, v, θ, ð, s, z, ʃ, ʒ, m, ŋ, l
o	t, d
i	ç, j

3 処理の概略

先に述べたように、英単語を片仮名表記するということは、英語の発音を日本語の音韻体系にあてはめて変形するということである。しかし英語の綴りは、文字と発音とが一對一に対応していないので、英単語の綴りから片仮名へ直接変換することは非常に困難である。そこで、英単語の綴りから片仮名へ直接変換するのではなく、まず綴りからその英語の発音へと変換し、そしてその英語の発音から日本語の読み、つまり日本語の発音に変換したのち、さらに片仮名に変換することにする。図3に大まかな処理の流れを示す。各段階では、置き換え規則を使って変換することにする。英語の発音は発音記号を用いて表記することにし、日本語の発音はローマ字を用いて表記することにする。例えば、

英単語 *cat* は、発音記号 /kæt/ にまず変換され、日本語の発音（ローマ字）*kyatto* を経て *キャット* に変換される。

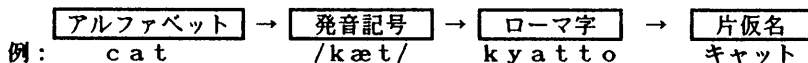


図3 処理の流れ

英語の綴りと発音は、一般に不規則であると考えられている。そのためアルファベットから発音記号への変換がかなり難しいものと予想される。発音記号からローマ字への変換は、かなり規則性があることがはっきりしているためアルファベットから発音記号への変換ほど難しくはないものと考えられる。最終段階の変換にあたるローマ字から片仮名への変換は、ローマ字仮名変換であるので、この変換については本稿では触れないことにする。

4 置き換え規則

置き換え規則は、一般に次の形をしている。

$$XAY \rightarrow XBY \quad (1)$$

これを生成音韻論の音韻規則の定式¹²⁾で表わすと次の通りである。

$$A \rightarrow B/X_Y \quad (2)$$

ここで、A, B, X, Y はいずれも文字列を示し、 \emptyset (ゼロ文字列) を含むことができる。この両式は「A が X の後にあり、かつ Y の前にあるとき、B に置き換える。」ということを表わす。A = \emptyset のときは、B ($\neq \emptyset$) の挿入規則となり、B = \emptyset のときは、A ($\neq \emptyset$) の削除規則となる。また A = B = \emptyset ということはありえない。

例えば、英単語の綴りでは「語末(単語のおわり)の e は、発音されない。」という規則¹⁶⁾があるが、これを置き換え規則として記述してみよう。この場合、語末を示すために記号 # を用いることにすると、上の式の各要素はそれぞれ、

$$A = e, B = \emptyset, X = \emptyset, Y = \#$$

となるので、式(1)の形式で書き表わすと、

$$e\# \rightarrow \#$$

となり、式(2)の形式で書き表わすと、

$$e \rightarrow \emptyset / _ \#$$

となる。

いくつかの置き換え規則をグループ化して一つの置き換え規則にまとめることができる。以下の二つの規則を考えてみる。

$$A \rightarrow B/X_Y \quad : R-\alpha$$

$$A \rightarrow B/Z_Y \quad : R-\beta$$

この両式を比較してみると、R- α の X と R- β の Z の部分以外はすべて同じである。この両式は、この X と Z の違い以外は同じなので次のような一つの規則としてグループ化することができる。

$$A \rightarrow B/\{X, Z\}_Y \quad : R-\gamma$$

ここで、括弧 { } は、この括弧でくくられ、コンマで区切られた各文字列のうちどれか一つを選択することを示すことにする。

英単語の綴りと発音との関係において、次のような規則がある。

「子音字 c は、母音字 e, i, y の前では /s/ と発音される。」¹⁷⁾

これを置き換え規則で表わすと、

$$c \rightarrow /s/ / _e \quad : R-1$$

$$c \rightarrow /s/ / _i \quad : R-2$$

$$c \rightarrow /s/ / _y \quad : R-3$$

という3つの置き換え規則となる。この3つの規則をグループ化して一つの置き換え規則にすると次

のようになる。

$$c \rightarrow /s/ / _ \{e, i, y\} \quad : R-4$$

グループ化では図4のような4つの基本的なパターンがある。図4(1)は変換される文字列(A, A')のグループ化、(2)は変換される文字列(A)の前条件の文字列(X, X')のグループ化、(3)は変換される文字列(A)の後条件の文字列(Y, Y')のグループ化、(4)は変換後の文字列(B, B')のグループ化である。最後のグループ化(4)は、文字列AをB, B'のうちどちらに置き換えるべきかはっきりしない。このように変換後の文字列のグループ化は、置き換え規則の式が曖昧になるので、ここでは扱わないことにする。

- (1) $A \rightarrow B / X_Y$
 $A' \rightarrow B / X_Y$ $\Leftrightarrow \{A, A'\} \rightarrow B / X_Y$
- (2) $A \rightarrow B / X_Y$
 $A \rightarrow B / X'_Y$ $\Leftrightarrow A \rightarrow B / \{X, X'\} _Y$
- (3) $A \rightarrow B / X_Y$
 $A \rightarrow B / X_Y'$ $\Leftrightarrow A \rightarrow B / X_ \{Y, Y'\}$
- (4) $A \rightarrow B / X_Y$
 $A \rightarrow B' / X_Y$ $\Leftrightarrow A \rightarrow \{B, B'\} / X_Y$

図4 グループ化の基本パターン

置き換え規則を使った変換では、規則を適用する順序が非常に重要である。例えば、cat という単語は次の3つの置き換え規則⁽⁶⁾を適用することによって発音記号に変換可能であるが、正しい順序で置き換え規則を適用しなければ正しく発音記号に変換されない(図5参照)。

$$c \rightarrow /k/ / _ a \quad : R-5$$

$$a \rightarrow /æ/ / _ t \# \quad : R-6$$

$$t \rightarrow /t/ \quad : R-7$$

図5(a)は上の3つの置き換え規則を R-5, R-6, R-7 の順に適用した場合の例で正しく変換される。一方、図5(b)は、R-5, R-7, R-6 の順で適用した場合の例で正しく変換されない。

$\# \underline{c} \underline{a} \underline{t} \#$ $\quad /k/ \quad : R-5$ $\quad \quad /æ/ \quad : R-6$ $\quad \quad \quad /t/ \quad : R-7$ $\# /k/ /æ/ /t/ \#$ $\quad \downarrow$ $\quad /kæ t/$ $\quad (a)$	$\# \underline{c} \underline{a} \underline{t} \#$ $\quad /k/ \quad : R-5$ $\quad \quad \quad /t/ \quad : R-7$ $\quad \quad \quad \quad : R-6$ $\# /k/ \underline{a} /t/ \#$ $\quad \downarrow$ $\quad * /k/a/t/$ $\quad (b)$
--	---

図5 適用順序の異なる変換例

変換規則の適用順序は現在のところ以下の場合以外は不明なので、経験的に決めている。

「規則は $XAY \rightarrow XBY$ のように表わされているが、ある規則 R- α の左辺が別の規則

R-β の左辺の部分列になっているとき、この二つの規則の適用順序を R-β, R-α とする。
 例えば、

o a → /ou/ : R-8
 o a r → /oər/ : R-9

のような場合、置き換え規則 R-8 の左辺 o a は、置き換え規則 R-9 の左辺 o a r の部分列になっているので、この二つの置き換え規則の適用順序は R-9, R-8 とする。

5 アルファベットから発音記号への変換

「3 処理の流れ」で述べたように変換の第1段階として英単語の綴り字から置き換え規則を用いてその語の発音を導出している。このとき発音の表記には発音記号が用いられている。

英語の発音を表記する発音記号（音声表記）には、いくつかの種類があるが、表記法は大別すると次に示す2種類の表記法^[2]となる。一つはできるだけ使用記号の数を少なくして同一記号に複数の解釈をしようとする D. Jones の「簡易表記」であり、もう一つはできるだけ正確に発音を表わすために、多くの記号を使って、誤った解釈が起きる余地をなくそうとする「精密表記」である。「簡易表記」であっても「精密表記」であっても、日本語の発音（ローマ字あるいは片仮名）に変換された時点では、ほとんど違いがないものと予想される。また日本では、ほとんどの英和辞典では D. Jones 式の「簡易表記」が採用されている。さらに「簡易表記」では使用する記号が少ない。以上の理由より、ここでは表4に示す D. Jones の「簡易表記」を変換に用いる発音記号として使うことにする。

表4 発音記号

子音	母音
閉鎖音 : p, b; t, d; k, g	単母音 : i, e, æ, ɔ, u, ʌ, ə
摩擦音 : f, v; θ, ð; s, z; ʃ, ʒ; h	長母音 : i:, a:, ɔ:, u:, ə:
破擦音 : tʃ, dʒ	二重母音 : e i, o u, a i, a u, ɔ i
鼻音 : m, n, ŋ	
流音 : l	
半母音 : r, j, w	

先に述べたように一般に英語の綴りと発音は不規則であると考えられている。しかし、英語の綴りは全くでたらめなものではなく綴りと発音との間に一定の規則が存在する。

イギリスやアメリカでは、このような綴りと発音との関係を学習するものに「フォニックス(phonics)」と呼ばれる学習法がある。ここでは、この「フォニックス」と呼ばれる学習法の、綴りと発音との間の規則を置き換え規則として考えてアルファベットから発音記号への変換を行う。

綴りと発音の関係には、同じ文字列が複数の発音に読まれる場合がある。例えば、綴り e a は /e/ あるいは /i:/ と発音される。このような場合「4 置き換え規則」で述べたようにグループ化すると式が曖昧になるので、式のうち最も多く使われると思われる方を規則として選び、残った方は例外として扱うことにする。綴り e a の場合には /i:/ と発音されることが多いので、これを規則とし、残った規則は例外とする。

e a → /e/ : 例外
 e a → /i:/ : 採用

6 発音記号からローマ字への変換

ここでの変換は、英語の音韻体系から日本語の音韻体系への転写、つまり英語の発音を日本語の発音に変換するものである。このとき日本語の発音の表記には、ローマ字を使用している。

ローマ字は、片仮名と同じく表音文字と呼ばれる。片仮名は、日本語の音韻体系の大きな特徴である「開音節」にあわせて一つの文字が「子音+母音」の組みを表わす。そのため発音記号から片仮名へ直接変換せずに、発音記号から、基本的に子音と母音を独立して表記するローマ字へ変換し、その後片仮名に変換するほうが良いものと考えられる。

現在広く使われているローマ字には、訓令式ローマ字とヘボン式ローマ字（正式には、標準式ローマ字）とがある。訓令式ローマ字では、「チ」を ti あるいは chi と表記し、英語でよく出てくる発音 /tʃi/, 片仮名で表わせば、「ティ」の音を表記する方法がない。一方、ヘボン式ローマ字は、子音の表記は英語の表記に似せて、母音の表記はイタリア語の表記に似せて作られているので、訓令式ローマ字では表記できない「ティ」や「トゥ」などの英語からの外来語特有の音を表記することができる。そのため、ここではヘボン式ローマ字を使って変換を行うことにする。

発音記号からローマ字へ変換する規則では、原音主義¹⁷⁾をとるべきか慣用主義¹⁷⁾をとるべきかという問題が生じる。原音主義とは、できるだけ英語の発音、つまり原音に近い形で忠実に表記しようとする考え方である。例えば、英語の発音 /vi/ をローマ字では vi と表記し、片仮名では「ヴィ」と表記して、bi、「ビ」と区別しようとする。実際には、この vi あるいは「ヴィ」は、bi あるいは「ビ」と同じ音で発音されている。最近の外来語では、この原音主義に立つものが多い。一方、慣用主義とは、外国語音を日本語の音韻体系に忠実にあてはめ、/vi/ のような音は、ローマ字では bi、片仮名では「ビ」と表記しようとする考え方である。また、比較的はやい時期に借用された語は、日本語化がはげしく原音とかなりかけはなれているものが多く、このような語も慣用主義に立つものとする。

ここでは、このような原音主義をとるか慣用主義をとるかは、「好み」の問題であるものとして考え、最近の外来語は原音主義に立つものが多いという現状を考慮して、原音主義を採用することにする。

発音記号からローマ字への変換は「2 英語と日本語の音韻体系」で述べた子音の対応、母音の対応、音節構造の違いによって添えられる母音などをアルファベットから発音記号への変換と同様に置き換え規則で表わして行う。

7 実験結果

まず、学習用英和辞典¹⁸⁾の a から e までの重要語のうち単音節語について変換を試みた。アルファベットから発音記号への変換は、「英語のフォニックス」¹⁹⁾を参考にして 96 の規則を使い、発音記号からローマ字への変換は、小泉の著書¹⁴⁾と松下の記事⁹⁾を参考にして 68 の規則を使って行った。その結果を表5に示す。総語数 483 のうち、正しく変換された語数は 359 であり、正答率は 74.3% である。正しく変換されなかった語のうち、新たに規則を付け加えることによって正しく変換され得る語（半例外）の数が 26、綴りと発音との間に（アルファベットから発音記号への変換において）規則性が見いだせない語（例外）の数が 98 である。発音記号からローマ字への変換については、すべての語について正しく変換された。

表5 変換結果

	語数	正変換語数	誤変換語数		正答率 (%)
			半例外	例外	
a	37	24	3	10	64.9
b	170	128	7	35	75.3
c	167	128	5	34	76.6
d	93	69	9	15	74.2
e	16	10	2	4	62.5
合計	483	359	26	98	74.3

次に、同じく学習用英和辞典の a から e までの重要語のうち多音節語について、単音節語と同様の規則を用いて変換を行った。その結果、変換を行った 1447 語全ての単語について正しく変換されなかった。この原因として、この変換方法では、音節とアクセントという概念を考慮していないので、母音が正しく変換されなかったことが挙げられる。

8 おわりに

置き換え規則を用いて英単語の綴りからその日本語音の片仮名へ変換する方法について述べた。この方法では、まず英単語の綴り（アルファベット）からその語の発音を表記する発音記号へ変換し、次にその発音記号からその日本語音を表わすローマ字へ変換して、最後にそのローマ字を片仮名に変換する。この3つの変換のうちアルファベットから発音記号への変換が他の2つの変換より難しいこと、単音節語では綴りとその発音との関係に規則性があれば良い結果が得られることがわかった。

今後の課題として、変換規則の適用順序および多音節語の変換方法の検討が挙げられる。

参考文献

- [1] 楊 震宇：談話室「外来語に見る音節の修正 - 日本語と中国語を比較して」, 言語生活, No.404, pp.78-79, 筑摩書房, 1985. 7
- [2] 島岡 丘, 佐藤 寧：最新の音声学・音韻論 - 現代英語を中心に -, 研究社, 1987
- [3] 竹林 滋：英語音声学入門, 大修館書店, 1982
- [4] 小泉 保：日本語の正書法, 大修館書店, 1978
- [5] 榎垣 実：日英比較語学入門, 大修館書店, 1976
- [6] 竹林 滋：英語のフォニックス, ジャパンタイムズ, 1988
- [7] 榎垣 実：日本外来語の研究, 研究社, 1963
- [8] 竹林 滋, 小島義郎：ライトハウス英和辞典, 研究社, 1986
- [9] 松下周二：連載「チャレンジコーナー, シニア版」, 言語, Vol.18 No.10, pp.119-121, 大修館書店, 1989. 10