Vol. 35 No. 1 情報処理学会論文誌 Jan. 1994

英語固有名詞の片カナ変換

住吉英樹† 相沢輝昭†

英日機械翻訳システムの翻訳結果の出力形態の一つとして，日本語音声合成装置による読み上げが考えられる．翻訳結果には，辞書に登録されていない英語固有名詞（人名，地名など）が，未知語として英語のまま含まれることがある．このような文章を日本語音声合成装置で読み上げさせると，英語部分の綴りをアルファベットのまま読み上げるため，非常に違和感を与えるだけでなく，文章の意味を容易に把握できない．この問題を解決するために，簡易な手法により英語固有名詞を片カナ読みに変換するプログラムを開発した．このプログラムは変換対象となる英語文字列中の，4文字の母音字子音字のパターンに注目した簡単なルールと，小規模な片カナ読みへの変換テーブルにより変換を行う．英語固有名詞（人名，地名）約1,500語のうち，80%以上が正しく変換でき，簡単な手法にしては高い変換精度を得ることができた．

Simple Translation of English Proper Nouns to Japanese Katakana (Phonogram Expression)

Hideki Sumiyoshi† and Teruaki Aizawa†

A Japanese speech synthesizer is a handy output device of an English-Japanese machine translation system. However, on encountering unidentified words, the translation system puts out English words in roman letters in the Japanese text, which are problematic to the speech synthesizer. A Japanese speech synthesizer usually reads out these words letter by letter in alphabet, which greatly degrades the quality making it difficult for a listener to comprehend what is spoken. Proper nouns such as human names and place names quite often fall in this case. In this paper, a method of converting English words into Japanese Katakana description is proposed. This enables a Japanese speech synthesizer to pronounce English as a word. The method includes simple rules and a small conversion table. They check four adjacent letters. The experiment on 1,500 English human names and place names showed correct conversion rate of more than 80%.

1. はじめに

英日機械翻訳システムの出力形態の一つとして，日本語音声合成器を用いて翻訳結果を読み上げ，結果の確認を容易にしたり，ヒューマンインタフェースを向上させたりすることは，実用上非常に有効である．

われわれが外電ニュースの翻訳などに利用しようと考えて開発している自動翻訳システムでも，日本語音声合成器の利用を考えている．ここで問題となるのが翻訳結果に含まれる英語部分の読み上げである．

ニュース文には，様々な人名や地名が頻繁に現れる．現在使用している英日機械翻訳システムでは，約17万語の大規模な辞書を利用しているが，人名や地名がすべて辞書に登録されているわけではない．辞書に登録されていない人名や地名などの英語固有名詞は，未知語として処理され翻訳結果の日本語の中に英語のままで出力されてしまう．一方，一般的な日本語音声合成器は，英語を英語の発音で読み上げる機能がなく，翻訳結果に含まれる英語部分は，綴りを一文字ずつアルファベットとして読み上げてしまう．このような合成音では，聞く者に違和感を与えるだけでなく，文章の意味の把握が困難になるなど，利用する際に大きな問題となる．この問題を解決するために，英語文字列から片カナ読みへの簡単な変換手法を考案した．

本手法は，翻訳結果（日本語）に含まれる未知語（英語）を対象とした，英語文字列から片カナ文字列への変換手法であり，変換の際に音素情報や規模の大きな辞書，多くの規則などを使用するのではなく，ごく少数の変換規則と小さなテーブルにより，英文字から片カナ文字へ，直接変換することを特徴としている．

† NHK 放送技術研究所先端制作技術研究部
NHK Science and Technical Research Laboratories

ドイツ語のように正書法（言葉の書き表し方）の規則が明確な場合には，綴りから発音への変換が容易であるが，英語には例外が多く，単純な規則での変換は容易ではないとされている．英語にも“phonics”と呼ばれる，綴りから発音を推測するための方法がある[1]．“phonics”では，基本的な文字，および文字群に対応した発音記号を設定することで，綴りから発音を推測することができる．文献1）では，定義した20の基本ルールにより，英和辞書（研究社「ライトハウス英和辞典」）の約2,600語以上を正しく発音することが可能であるとしている．

しかし，“phonics”では，文字の組み合わせによる発音を発音記号などにより定義しており，文字（英語）と文字（片カナ）の対応を規定しているわけではない．また，文字列への区切りは，人が行うことを前提にしており，ルールとして定義されていないため，計算機での自動処理を行うことができない．発音に関する20の基本ルールについても，複数の発音記号と複数の文字を組み合わせた定義が数多くあり，相当複雑な処理になることが予想される．

また，英語文字列から英語の音素への変換については，英語の音声合成器への適用を目的として，数多くの研究がなされている[2]．

英語の音声合成器の代表的な商用システムであるDECTalkとその研究システムであるKlatTalk[3]では，辞書および文法を用いており，辞書にない場合は約500の文字—音素間の変換ルールにより音素を生成する．

NetTalk[2]では，文字列と音素との対応をニューラルネットを利用して学習させている．学習には，20,000語の音素辞書を用いている．

これらのシステムで使用されている変換手法は，変換率が高く，正しい音素を生成できると報告されているが，かなり大きな発音辞書と多くの変換ルールを利用する必要がある．

英語まじりの日本語文を読み上げることを考えた場合，上記のような英語の規則合成手法を用いた音声合成器を翻訳結果の英語部分にのみ適用することも考えられるが，この場合には日本語の発音のなかに突然英語特有の発音が現れ，かえって文章の理解を困難にしてしまうなどの欠点がある．

このように，DECTalkやNetTalkで用いられている手法を本論文で目的としている，日本語音声合成器を利用した英語まじりの日本語文章の読み上げに応用するためには，かなりの修正が必要であると考える．

英語の綴りから片カナへの変換手法については，文献4）がある．これは英語の音素と日本語の音素との対応により変換を行うもので，英語綴り→英語発音記号→日本語発音記号→日本語片カナと変換をしている．この手法では，英和辞書（研究社「ライトハウス英和辞典」）の重要語483語に対する変換実験で，単音節語に対しては74.3%と高い正答率が得られるが，多音節語については，すべての単語が正しく変換できなかったと報告されており，変換精度は低い．

われわれは，英語まじりの日本語文章を日本語音声合成器で読み上げさせることを前提に，複雑で処理の規模が大きくなりがちな発音辞書や，多くの規則を用いる従来の手法とは異なる簡易な手法をとることにした．

変換対象となる英語文字列の4文字を単位とした母音字，子音字の並び方によって，片カナ1音に対応する文字群へ切り出しを行い，切り出した文字群をさらに子音の要素と母音の要素に分解し，小規模なテーブルを参照することで片カナに変換する手法である．

簡易な処理のため翻訳結果中に含まれる英語固有名詞を片カナ読みにリアルタイムで変換することができる．この手法を用いれば簡単な処理で，高い変換精度を得ることができ，翻訳結果を日本語音声合成器で読み上げる時の違和感を大幅に減少できた．

本論文では，2章で変換プログラムの処理内容を，3章で変換プログラムを用いた変換実験および実験結果について述べる．

2. 変換処理

2.1 基本的な考え方

入力となる英語文字列から，片カナ読みの1音節に対応する文字列へ切り出し，さらにその文字列を子音と母音に相当する文字に分解し，あらかじめ用意した片カナへの変換テーブルから対応する片カナを導く，というのが本方式の基本的な考え方である．

変換処理は，入力英単語の子音字と母音字の並び方に注目して，片カナ読みの1音節に対応する子音字と母音字の組み（1文字〜4文字）を切り出すプログラムと，切り出した文字群からさらに分解した子音字と母音字によって検索できる変換テーブルで構成している．**図1**に変換処理概念図を示す．

2.2 子音字，母音字の定義

本手法では，子音字（C）および母音字（V）について，以下のように定義した．

Type C1：1文字子音字

［b］，［c］など1文字で子音となる文字．

Type C2：2文字子音字

[ph]，[th] など二つの子音字が連続して一つの子音となる文字の組み．

Type V1：1文字母音字

［a］，［i］など1文字で母音となる文字．

Type V2：2文字母音字

[er]，[ir] など2文字連続することで，一つの母音要素となる文字の組み．

Type V3：3文字母音字

[eer]，[our] など3文字連続することで，一つの母音要素となる文字の組み．

2.3 片カナ読み変換テーブル

片カナ読みへの変換テーブルは，基本的には2.2節で定義した子音字，母音字により構成し，子音字群を列のエントリーに，母音字群を行のエントリーとする，2次元配列である．

変換テーブルの列エントリーには，2.2節で定義したC1，C2の子音字以外にV1の1文字母音字も含めている．また，行エントリーにも組となる母音字がない場合（単独変換）の読みを，特別な行として含めている．この理由を以下に述べる．

本方式での標準的なカナへの変換は，子音字と母音字の組によりテーブルの検索を行うが，組となる母音字がなく，どちらか単独でカナに変換しなければならない場合（単独変換）がある．

例えば，[Africa] という文字列を対象とした場合を考えると，片カナの音節に対応する文字への切り分けは［A］［f］[ri] [ca] のように行われる．ここで母音字［A］については1文字で［ア］に，子音字［f］についても1文字で［フ］に変換する必要がある．そこで切り出しを行った結果が，組となる母音字がない場合，つまり子音字もしくは1文字母音字単独になった場合でも片カナ読みへの変換を可能とするため，テーブルに単独変換のための行エントリーを設け，1文字母音字を含む列エントリーから，単独変換用行エントリーへの参照を行っている．これにより本手法では，どのような入力文字列の組み合わせに対しても必ず片カナ文字列を生成することができる．

また，[easton] などの場合には先頭の［ea］は，［イー］と変換するほうが正しいと考えられるので，1文字母音字+1文字母音字を一つの片カナ音節として扱うことにした．

このように母音字を子音字のように扱ったほうが正しい読みが得られる場合があるので，列エントリーに1文字母音字を含めている．

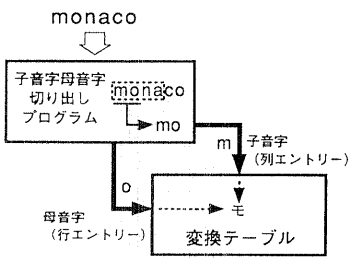


図1 変換処理概念図
Fig. 1 The process image.

表1 子音字
Table 1 Defined consonants.

子音字（列のエントリー）			
1文字子音		2文字子音	
A	N	CH	LL
B	O	SH	MM
C	P	PH	NN
D	Q	GH	PP
E	R	TH	RR
F	S	WH	SS
G	T	CK	TT
H	U	BB	ZZ
I	V	CC	KN
J	W	DD	DG
K	X	FF	
L	Y	GG	
M	Z	KK	

表2 母音字
Table 2 Defined vowels.

母音字（行のエントリー）				
1文字母音字	2文字母音字		3文字母音字	単独変換
A	AR	EI	EER	*
I	IR	OI	IEW	
U	UR	IA	EOU	
E	ER	IU	ARE	
O	OR	IE	OUR	
Y	AU	IO	EAR	
	UA	OU		
	UE	OO		
	EA	OY		
	EE	OW		
	AI	EW		
	II			
	UI			

なお，子音字と同様に1文字母音字を扱うのは，変換テーブルにおいてのみであり，切り出しの文字数を決める母音字，子音字の並び方パターン判別の際には，当然ながら区別して扱う．

変換テーブルに用意した子音字，母音字の種類と数を以下に示す．また，子音字の種類を**表1**に，母音字の種類を**表2**に示す．

列のエントリー	**…(49個)**
1文字子音字（C1）	20個
1文字母音字（V1）	6個
2文字子音字（C2）	23個
行のエントリー	**…(37個)**
1文字母音字（V1）	6個
2文字母音字（V2）	24個
3文字母音字（V3）	6個
単独変換	1個

表2の［*］は，単独変換のための行エントリーを示している．

テーブルに登録しておく片カナが変換結果に大きく影響するので，変換テーブルを作成するにあたっては，[va] に対応する読みを［ヴァ］とせず［バ］として登録するなど，英語の一般的な日本語表記に近いものをテーブルに入れ，より自然な片カナ表記に変換できるとともに，日本語音声合成器での利用を考慮して作成した．

テーブルサイズは，計算上 49 列×37 行＝1,813 個となるが，子音字と母音字の組にする際に4文字しか注目していないので，2文字子音字と3文字母音字の組み合わせとなる部分 (23列×6行＝138) は使用しておらず，1,813−138＝1,675 個となる．2.4 節で詳しく述べるが，2文字子音字と3文字母音字の組み合わせの出現頻度は高くないので，この制限は，変換率にほとんど影響を与えていない．

変換テーブルは，メンテナンスの便を考慮して，パーソナルコンピュータの表作成ソフトにより，作成，変更が行えるようにした．

変換テーブルの概念を与えるため，**図2**に変換テーブルの概念図を，**表3**にテーブルの一部を示す．

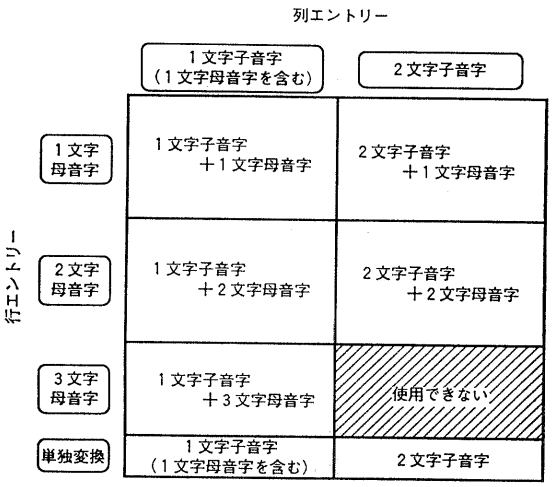


図2 片カナ読み変換テーブル概念図
Fig. 2 The conversion table image.

表3 変換テーブル（一部）
Table 3 The conversion table (part).

母音字＼子音字	a	b	c	d	e	f	g	h	i	j	k
a	オ	バ	キャ	デ	エア	ファ	ガ	ハ	イア	ジャ	カ
i	エイ	ビ	シ	ディ	イイ	フィ	ジ	ヒ	イー	ジ	キ
u	オー	バ	カ	デュ	ヨー	ファ	ガ	フ	イウ	ジュ	ク
e	アエ	ビ	ス	デ	エ	フェ	ジ	ハ	イー	ジェ	キ
o	アオ	ボ	コ	ドウ	エオ	フォウ	ゴー	ホ	イオ	ジョ	コウ
y	エイ	バイ	シィ	ディ	イィ	フィ	ジィ	ヒィ	イィ	ジィ	キィ
ar	ウォー	バー	カー	ダー	アー	ファー	ガー	ハー	イアー	ジャー	カー
ir	エアー	バー	サー	ディア	エイ	ファー	ガー	ヒア	イアー	ジア	カー
ur	アウア	バー	カー	デュア	ヨーロ	フア	ギュア	ハー	イウア	ジュー	カー
er	エアロ	バー	サー	ダー	イエー	ファー	ガー	ハー	イエア	ジェア	カー
or	オウア	ボー	コー	ドー	エオー	フォー	ゴー	ホー	イオア	ジョー	コー
au	アウ	ボウ	カウ	ファウ	オウ	ファウ	ガウ	ハウ	イアウ	ジャウ	カウ
ua	アウア	ブア	クア	デュア	エウア	フア	グア	フア	イウア	ジュア	クア
ue	アウエ	ブエ	クエ	デュウ	エウエ	フエ	グ	フエ	イウエ	ジュエ	クエ
ea	アエア	ビー	セア	ディ	イーア	フェア	ゲア	ヘ	イエア	ジー	キー
ee	アイー	ビー	シー	ディー	エエ	フィー	ギー	ヘー	イー	ジェー	キー
ai	エイ	バイ	ク	ダイ	エ	ファイ	ガイ	ハイ	イ	ジャイ	カイ
ii	アイー	ビー	シー	ディー	エイー	フィー	ギー	ヒー	イー	ジー	キー

2.4 カナ1音に対応する文字の切り出し

入力の英語文字列から，片カナ読みの1音節に対応する文字列の切り出し方については，いろいろな方法が考えられるが，本手法では入力となる英語文字列の4文字に注目し，2.2節で定義した子音字（C）と母音字（V）の並び方により，基本的な切り出し文字数を決め文字を切り出している．注目する文字数を4としたときの子音字，母音字の並び方のパターンは $2^4=16$ となる．以下に並び方のパターン，および括弧内に片カナの1音節として判断する基本文字数を示す．

VVVV(2) VVVC(2) VVCV(2) VVCC(2)
VCVV(1) VCVC(1) VCCV(1) VCCC(1)
CVVV(2) CVVC(2) CVCV(2) CVCC(2)
CCVV(1) CCVC(1) CCCV(1) CCCC(1)

切り出す文字数は，上記の基本文字数だけでは決められない．それぞれのパターンの中で2文字子音字や2文字，3文字母音字などを含む例外的な読みに対応する必要があるので，各パターンのなかで個別に例外処理（2.5節で詳しく述べる）を行い，それぞれ必要な文字数を切り出している．

連続子音字，連続母音字に対する処理の種類などから，いくつかのパターンで処理を共通化することができるので，**表4**に示す8種類となる．連続子音字，連続母音字などに対応する例外処理は，(例外)として示し，[ ]内には，片カナ1音節に対応する文字列として採用する文字数を示した．

連続子音字，連続母音字に対する具体的な例外処理については，2.5節で述べる．

切り出しのために注目する文字数を決めるために，以下のような検討を行った．

2.2節で定義した子音字（C1，C2），母音字（V1，V2，V3）から片カナ1音を構成する文字の組み合わせパターンは，以下の7種類が考えられる．なお，1文字子音字には1文字母音字を含む．

P1：子音字または1文字母音字 (49組)
P2：1文字子音字と1文字母音字 (156組)
P3：1文字子音字と2文字母音字 (624組)
P4：1文字子音字と3文字母音字 (156組)
P5：2文字子音字と1文字母音字 (138組)
P6：2文字子音字と2文字母音字 (552組)
P7：2文字子音字と3文字母音字 (138組)

これらの文字の組み合わせに切り出すために，注目する文字の数を多くすれば，変換精度が向上することが予想されるが，処理が複雑になるとともに，変換テーブルの規模が大きくなるなどの問題がある．

どのような文字の組み合わせが実際の英語の中で使用されているのかを知るために，英語辞書の片カナ訳語を持つ人名，地名データを基に各パターンの出現率を求めた．

調査したデータは，英和辞書（三省堂「ニューズ英語辞典」，小学館「最新英語情報辞典」）の中から片カナ訳語を持つ人名，地名を抜き出した（地名データ934語，人名データ1,018語）1,952語である．

2.2節で定義したそれぞれの子音字と母音字を組み合わせた文字列を，どの位の語が含んでいるか調べた．結果を以下に示す．子音字および1文字母音字単独での出現回数は省いた．

地名（延べ総数は 3,437）
P2 2,638/3,437 76.75%

表4 子音字，母音字のパターンと対応する処理
Table 4 The process types and consonants vowels pattern.

【処理：a】	VVVV，VVVC，VVCV，VVCC
(基本)	・1文字母音字＋1文字母音字［2文字採用］
【処理：b】	VCVV，VCVC，VCCV，VCCC
(基本)	・単独変換（1文字母音字）［1文字採用］
【処理：c】	CVVV
(例外)	・1文字子音字＋3文字母音字(eou)［4文字採用］または
(例外)	・1文字子音字＋2文字母音字［3文字採用］または
(基本)	・1文字子音字＋1文字母音字［2文字採用］
【処理：d】	CVVC
(例外)	・1文字子音字＋3文字母音字(eer,iew,our,ear)［4文字採用］または
(例外)	・1文字子音字＋2文字母音字［3文字採用］または
(基本)	・1文字子音字＋1文字母音字［2文字採用］または
【処理：d】	CVCV
(例外)	・1文字子音字＋3文字母音字(are)［4文字採用］または
(例外)	・1文字子音字＋2文字母音字［3文字採用］または
(例外)	・?a?e，?i?eの例外処理［2文字採用］または
(基本)	・1文字子音字＋1文字母音字［2文字採用］
【処理：e】	CVCC
(例外)	・1文字子音字＋2文字母音字(ar,ir,ur,er,or,oy,ow,ew)［3文字採用］または
(基本)	・1文字子音字＋1文字母音字［2文字採用］
【処理：f】	CCVV，CCVC
(例外)	・2文字子音字＋2文字母音字［4文字採用］または
(例外)	・2文字子音字＋1文字母音字［3文字採用］または
(基本)	・単独変換（1文字子音字）［2文字採用］
【処理：g】	CCCV，CCCC
(例外)	・単独変換（2文字子音字）［2文字採用］または
(基本)	・単独変換（1文字子音字）［1文字採用］

P3	585/3,437	17.02%
P4	10/3,437	0.29%
P5	167/3,437	4.86%
P6	35/3,437	1.02%
P7	2/3,437	0.06%

人名（延べ総数は 3,298）

P2	2,398/3,298	72.71%
P3	579/3,298	17.56%
P4	11/3,298	0.33%
P5	244/3,298	7.40%
P6	66/3,298	2.00%
P7	0/3,298	0%

上記のデータから，3文字以下の組み合わせ（P2, P3, P5）で約 97% 以上，4文字以下の組み合わせ（P2, P3, P4, P5, P6）で約99% 以上を占めていることがわかる．

出現する文字の組み合わせの被覆率からいえば3文字に注目することで十分と考えられる．しかし，3文字に注目した場合，2文字母音字を含む3文字目の切り出し時に以下のような問題が生じる．

3文字目までは同じ [car] という文字列だがこれに続く文字によって，3文字目の［r］をどう処理するか違ってくる．

3文字で切り分け

[carney]＝CVC（正しい切り分け）
＝[car] [ne] [y]
[carole]＝CVC（誤った切り分け）
＝[car] [o] [le]
（正しくは [ca] [ro] [le]）

もちろん例外的な発音をするものもあるので，すべてが正しく切り分けられるわけではないが，この場合には，4文字目が子音字であるか，母音子であるかを判断することで，切り分けが正しく行われる．

4文字で切り分け

[carney]＝CVCC
＝[car] [ne] [y]
[carole]＝CVCV
＝[ca] [ro] [le]）

この場合のように，2文字母音字の2文字目が子音字［r］や［w］，子音字的要素を持った母音字［y］のときに，切り出しに4文字目の情報が必要になる．これらの2文字母音字を含む語数を，上記と同様に辞書データに現れる回数を調べると，地名で約 8.6% (295/3,437)，人名で約 13.0%(431/3,298)あることがわかった．組となる子音字が2文字子音字の場合には5文字目の情報が必要となるが，このような組み合わせは，地名で約 0.4% (15/3,437)，人名で約 1.6% (53/3,298) と多くはない．

注目する文字を5文字にした場合には，2文字子音字と3文字母音字との組み合わせが変換できるが，この2文字の組み合わせ「P7」の出現頻度を上記の辞書データで同様に調査した結果，2語に過ぎないことがわかった．また，現在はテーブルにない3文字子音字として [tch], [ght], [cqu]，4文字母音字 [ower] を加えることができるが，これら3文字子音字，3文字母音字，4文字母音字と2.2節で定義した母音字，子音字群との組み合わせを含む語は4語でしかない．

これらのデータから［2文字子音字＋3文字母音字］，［3文字子音字＋2文字母音字］，［3文字子音字＋3文字母音字］の5文字以上の組み合わせにより一つの音節を作る組み合わせの出現頻度は高くなく，注目文字数を5文字にしても変換率を格段に向上させることは難しいと考える．一方で変換テーブルのサイズは，52×38＝1,976 へ拡大する必要がある．

さらに，片カナ1音節への切り出し精度を上げる目的で注目語数を5文字に増やした場合，母音字，子音字の出現パターンの組み合わせは，$2^5=32$ パターンとなる．いくつかのパターンで処理の共通化は可能と思われるが，処理の複雑さの増大が著しい．

以上の調査とともに，英日翻訳システムの翻訳結果に未知語として含まれている英単語についても検討を加え，切り出しに用いる文字を4文字にすることで片カナ一音に対応する組み合わせのほとんどが切り出せると判断し，本手法では注目文字数を4に決定した．

2.5 例外処理

子音字と母音字のパターンから採用する文字数は，基本的に2.4節で定義した文字数により切り出しを行うが，英語特有の綴りに対応させるための例外として，以下の（1）～（4）までの処理を行っている．

略語への対応は，すべての処理に先立って行うが，その他の例外処理は，**表4**に（**例外**）として示すように，各子音字母音字のパターンに対応した処理に組み込まれている．

（1）略語処理

変換処理を行う前には，入力英語文字列の先頭から2文字を調べ，小文字以外の文字が連続する場合，変換処理を行わず，入力文字列をそのまま出力する．これは英語ニュースに比較的多く含まれる略語への対応

で，略号をアルファベットで一文字ずつ読み上げさせるためである．

例：[CIS] [EC]

（2）連続母音子の処理

注目した4文字の子音字と母音字のパターンが，CVVV, CVVC, CVCV, CVCC, CCVC のパターンのときに4文字のうち，2, 3文字目または2, 3, 4文字目を調べ，2.2 節で定義した連続母音字の場合には，連続母音字に対する処理を行う．

ここで CVCV, CVCC, CCVC の場合にも連続母音字の対象に含めているのは，子音字Cの中に［r］など，子音字であるが［ar］のように母音字の後ろに続くことで単独の音としては発音されず，連続することで一つの母音の役目をするものがあり，本手法ではこれらを2文字母音字，3文字母音字として定義しているためである．

CVCV の場合は注目している4文字中の2文字目から4文字目までを，CVCC の場合は2文字目と3文字目までを，それぞれ文字単位でチェックし，変換テーブルの行エントリーにある2文字母音字，3文字母音字に該当する場合には，1文字目の子音字と対象となる2文字母音字，3文字母音字とで片カナ変換テーブルを検索する．2文字母音字，3文字母音字に該当しない場合に，1文字母音字を利用してテーブルを検索する．

例：[parker]

park＝CVCC

＝[p]+[ar]

（1文字子音字＋2文字母音字）

同様に CCVC パターンの場合には，先頭の2文字について2文字子音字のチェックを行い，2文字子音の場合には，3文字目と4文字目の文字が2文字母音字に該当するかチェックし2文字子音字＋2文字母音字で片カナ変換テーブルを検索する．

例：[charlie]

char＝CCVC

＝[ch]+[ar]

（2文字子音字＋2文字母音字）

2文字母音字に該当しない場合には，先頭2文字の子音字と3文字目の母音字でテーブルを検索する．

2文字連続母音字の場合の処理を**図3**に示す．

（3）2文字子音字の処理

2文字子音字への対応は，子音字，母音字のパターン判断で，子音が2文字連続するパターンと判断したとき，つまり，CCVV, CCVC, CCCV, CCCC のどれかのパターンに合致する時には，まず2文字子音字のチェックを行い，2.2 節で定義した2文字子音字の場合には，該当する2文字子音字から変換テーブルを検索する．それ以外の場合は，先頭の1文字を採用して，あらかじめ登録してある1文字単独での片カナ読みを変換テーブルから出力する．

例：[spark]

spar＝CCVC

＝[s]

（1文字子音字，単独変換）

[child]

child＝CCVC

＝[ch]+[i]

（2文字子音字＋1文字母音字）

2文字連続子音の場合の処理を**図4**に示す．

（4）[make], [like] などの処理

[make], [like] に代表される．[?a?e]（?は任意の子音字を表す）もしくは [?i?e] で構成される文字列を変換する場合，通常であれば [ma]=[マ] と変換するところを，[ma]=[メイ] と変換するような例外処理を行っている．

子音字，母音字のパターンが CVCV で，入力文字列が [?a?e]（?は任意の子音字を表す）もしくは [?i?e] のときにこの例外処理を行う．

2.6 処理手順

実際の変換処理は，以下の手順で行っている．処理手順の例を**図5**に示す．

（1）入力文字列の先頭から2文字を調べ，2文字とも英小文字以外であれば略号と判断し，入力文字列

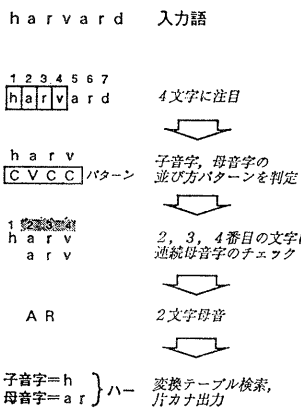


図3 2文字母音字の処理

Fig. 3 The process flow of the double vowel.

全体を出力として処理を終える．

（2）切り出しのために注目する4文字の，先頭位置を示すポインターの値を初期値＝[1]とする

（3）ポインターが指し示す文字から，4文字を取り出す．

（4）取り出した4文字について，あらかじめ定義している子音字か母音字かを調べ，その並び方を16パターンの中から選択する．

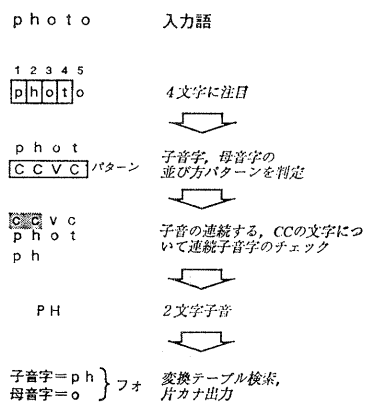


図4 2文字子音字の処理
Fig. 4 The process flow of the double consonant.

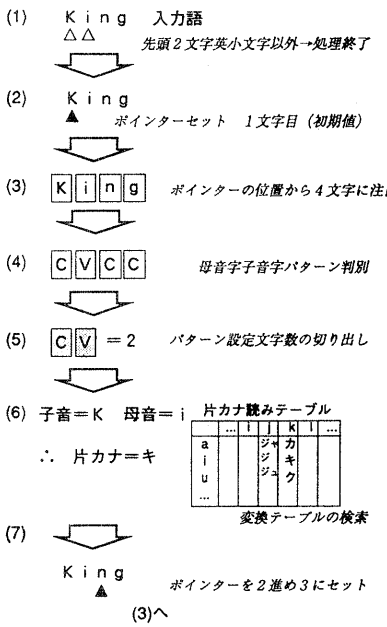


図5 処理手順
Fig. 5 The overall process.

（5）子音字，母音字の並び方の各パターンごとに，**表4**に示す各処理を行う．

（6）切り出した文字の子音字と母音字の組み合わせ（単独変換の場合は専用の行エントリー）から，片カナテーブルを参照し，テーブルに登録されている片カナを変換結果として出力する．

（7）（5）の処理によって切り出した文字数だけポインターを進め，注目する位置をずらして(3)に戻り，文字列の終わりまで上記処理を繰り返す．

なお，文字列の終わりの部分で，子音字，母音字パターンの判断に必要な文字が不足する場合は，文字のない部分を子音字と仮定して変換処理を進めている．

これは子音字か母音字のどちらかの情報がなければ切り出し処理ができないことと，子音字に設定しておけば，存在する文字だけで読みが出力できることからである．

3. 変換実験

変換プログラムはワークステーション上のC言語で記述しており，プログラムサイズは，約600ステップである．3 MIPS のワークステーションで，1秒間に300 語以上の高速な変換処理が可能である．

開発した変換プログラムの変換精度を測るため変換実験を行った．

翻訳システムで未知語として処理されるものは，圧倒的に人名，地名が多いため，テストデータは英和辞書（三省堂「ニューズ英語辞典」，小学館「最新英語情報辞典」）の中から片カナ訳語を持つ人名，地名を抜き出した．

片カナ訳語を持つ語のなかには，訳語として読みではなく通称や，略語の意味に相当する訳語が登録されているものがあった．これらを実験の元データとしたこでは正しい評価が得られないので，評価対象から除外している．以下に除外したデータの内容，語数と例を示す．

（1）略語の意味や日本での通称を訳語として登録されたもの

人名：2語 地名：90語

例：ariz：アリゾナ
ark：アーカンソー
ivory coast：コートジボアール

（2）フランス語など非英語系の地名，人名を，日本語での一般的な読みや，通称で訳語としてを登録されているもの

人名：119語　地名：143語

例：banear shahpour：バンダルシャープール
marseilles：マルセーユ
chun doo hwan：チョンドゥホアン
dietrich：デートリッヒ
de cuellar：デクエアル

これらの語を除いた，人名882語，地名678語，合計1,560語を変換実験のためのテストデータとした．

3.1 正変換率

開発した変換プログラムにより変換処理を行い，変換結果から正答率を求めた．片カナで外来語を表記する場合，大きな揺れを生じることが知られている．例えば，「コンピュータ」，[コンピューター]など，どちらの表記が正しいとも判断できない場合が多い．

片カナ語の表記の揺れに関しては，以下のような報告がある．

文献5）では日本語文章中に現れる片カナ語の表記の揺れを吸収する，異表記変換処理手法を提案している．例えば，「クに拗音を付加したものはカ列の各文字になる」などのルールにより，片カナの異表記を統一した表記に変換している．

文献6）では，発音をもとにした原表記とカナ表記の対応を判定するアルゴリズムについての提案がなされている．カナ表記から日本語音素列を生成する際に，2重母音の長音化，母音にはさまれた長音の削除などの規則により片カナ語を統一表記化し日本語音素列を生成している．

文献7）では片カナ語から原語（英語）の対応付けを行うために，片カナの音韻的性質に基づいて原語を推定する手法を提案している．「[ア]に対応する原語として[a]，[u]，[ar]，[ya]」など総数900の変換規則により，片カナ文字列から推定される複数の原語を生成する実験を行い，実験した全語数2,814のうち，候補が得られたものは95.8%（2,695語），うち正解が含まれたものは，79.4%（2,235語），第一候補が正解だったものは74.7%（2,101語）と報告されている．

本論文では，これらの文献で述べられている片カナの表記の揺れを参考にし，日本語音声合成器での利用を前提とした変換結果の評価を行った．具体的には，実験結果のうち以下の(1)～(3)について正しく変換されたものとした．

（1）完全一致のもの

変換処理結果と辞書に登録されている片カナ訳語の一致を比較した．完全に一致したものは以下の語数であった．

人名237語／882語　26.9%
地名117語／678語　17.3%

（2）一文字違いのもの

辞書に登録されている訳語との違いが1文字である語について内容を検討した．1文字違いのものには，以下のようなパターンがある．

（a）同一文字数で1文字違い
（b）文字数が1文字長く，1文字違い
（c）文字数が1文字短く，1文字違い

これらのほとんどが[ティ]と[テー]，[ショー]と[ショウ]に代表されるような，片カナでの表記の違いによるものであった．この違いは，音声合成器による読み上げでは，聞き分けることができず，実用上問題がないので，誤差1文字以内の語を，正しい変換とした．該当する語数は以下のとおり．

人名232語／882語　26.3%
地名185語／678語　27.3%

（3）2文字以上違うもの

2文字以上違うものについても，文献5）で述べられている片カナ表記の揺れを参考に，日本語音声合成器での合成音の聞き取りにおいて実際上問題にならないものを正変換とした．例えばedison（辞書訳語：エジソン）を[エディソン]，kingsley（辞書訳語：キングズレー）を[キングスレイ]など．語数は以下のとおり．

246語／882語　27.9%
256語／678語　37.7%

（4）正変換率

以上の(1)～(3)を正変換とすると，全体で次の正変換率が得られた．

人名：715語／882語＝81.1%
地名：558語／678語＝82.3%
全体：1,273語／1,560語＝81.6%

簡易な変換手法にもかかわらず，80%を越える高い変換精度が得られている．

3.2 誤変換の要因

一方，誤変換と判断した287語について，誤変換の要因による分類を行った．結果を以下に示す．

（a）変換テーブル登録片カナの不適切

nevada（辞書訳語：ネバダ）　[ネバデ]
cairo（辞書訳語：カイロ）　[ケイロ]
202語　70.3%

(b) 片カナに対応する文字の切り出し失敗
salisbury (辞書訳語：ソールズベリー)
[サリスバーウィ]
karen (辞書訳語：カレン) [カーン]
(3文字母音字 [are])
35語 12.2%

(c) 該当母音字なし
john (辞書訳語：ジョン) [ジョヒン]
19語 6.6%

(d) 該当子音字なし
write (辞書訳語：ライト) [ウレイト]
23語 8.0%

(e) 例外処理の誤用
belize (辞書訳語：ベリーズ) [ベライズ]
(?i?e) の誤用
2語 0.6%

(f) その他
mcQueen (辞書訳語：マックイーン)
[語の3文字目の大文字により変換失敗]
9語 3.1%
(一語中に複数の誤変換個所あり)

上記の(a)に示すように，変換テーブルに登録された不適切な片カナ読みによって，誤変換となったものが約 70% と最大要因となっており，変換テーブルの改善に余地を残している．この中には，単純にテーブルの修正で正しい結果が得られるものと，テーブルの修正だけでは対応できない．同じ綴りに対して複数の発音を持つものがある．このようなものには，例外的な処理の追加，もしくは辞書機能を付加して対応することが考えられるが，簡易な処理を目的とする本手法では，出現頻度の高いものをテーブルに登録する方法で対応することを検討している．

その他，切り出しの失敗の要因として，該当する子音字，母音字を定義していないものがいくつかあった．

また，切り出しに失敗した要因の中には，(a)と同様に [ter] を，[ター] と読む場合と [テル] と読みわける必要があるなど，例外処理を行うか，辞書機能を付加しなければ，一つの処理パターンで正しく切り出すのは困難なものもある．

4. おわりに

本手法では英日翻訳システム用の辞書にも登録されていない未知語（比較的語長が長い語が多い）の簡易な変換を目的とすることで，同じ子音字，母音字の組み合わせによる読みが複数ある場合に正しく処理できない点など，簡易な手法の限界はあるものの，テストデータの 80% 以上が正しく変換された．

変換処理プログラムを通した翻訳結果は，翻訳システムの辞書に登録されていない未知語も，その大部分が読み上げに問題のない片カナ読みに変換され，日本語音声合成器により，普通に読み上げることができた．これにより理解度が大幅に向上し，翻訳結果の出力形態として有効に利用できることが確認できた．

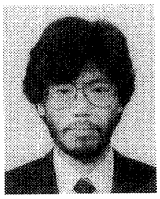
今後は，今回の実験結果を基に，人名，地名に対象を絞った変換テーブルの修正，および変換アルゴリズムの最適化，例外処理の強化など検討しているが，誤変換語の中の，同じ子音字，母音字で別の読み方をするものについては，各子音字，母音字の出現回数の高いものから変換テーブルのエントリーとして採用するなど，変換テーブルや例外処理部をあまり大きくしない方法で対応し，簡易な処理のままで変換精度を向上させることを検討したい．

謝辞 本研究を進めるにあたってご指導いただいた NHK 放送技術研究所 先端制作技術研究部の二宮部長，ならびに江原主任研究員をはじめとする自動翻訳グループ各位に感謝します．

参考文献

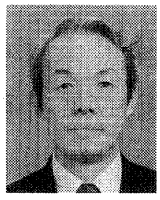
1) 竹林 滋：英語のフォニックス，ジャパンタイムズ (1981).
2) Klatt, D. H.: Review of Text-to-speech Conversion for English, *J. Acoust. Soc. Am.*, Vol. 82, No. 3, pp. 737-793 (1987).
3) Klatt, D. H.: The Klattalk Text-to-speech Conversion System, *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process*, Vol. 1982, No. Vol. 3, pp. 1589-1592 (1982).
4) 堀内雄一ほか：英単語のアルファベット表記から仮名表記への変換，情報処理学会自然言語処理研究会報告，No. 79-1, pp. 1-8 (1990).
5) 伍井ほか：カタカナ異表記処理，第38回情報処理学会全国大会論文集，4E-2, pp. 351-352 (1989).
6) 大深：原表記とカナ表記の対応判定アルゴリズム，第37回情報処理学会全国大会論文集，6B-3, pp. 1010-1011 (1988).
7) 野美山：カナタカナ外来語の表記の揺れの解消，第41回情報処理学会全国大会論文集，6S-3, pp. 191-192 (1990).

(平成5年1月25日受付)
(平成5年10月14日採録)



住吉 英樹

1961年生．1980年広島県立広島工業高校電気科卒業．同年NHK入局．広島放送局を経て1984年から放送技術研究所において計算機利用技術，ヒューマンインタフェースの研究に従事．現在，先端制作技術部研究部に所属．テレビジョン学会会員．



相沢 輝昭（正会員）

1940年生．1963年京都大学工学部電気工学科卒業．同年NHK入局．放送技術研究所において日本語処理，データベース放送ニュース用英日機械翻訳等の研究に従事．現在，放送技術研究所主幹研究員．電子情報通信学会，ACL各会員．