An Algorithm for Generating a Dictionary of Japanese Scientific Terms

YUN-SUN KANG and ANTHONY A. MACIEJEWSKI
Purdue University, West Lafayette, USA

Abstract

This article describes the implementation of algorithms for generating a dictionary of Japanese scientific terms originating from the English language. Such words are typically transliterated into *katakana*, one of the four distinct orthographies that commonly occur in Japanese texts. The effort required to learn *katakana* yields significant returns to readers of technical Japanese due to the high incidence of terms derived from English. The algorithms described here are able to automatically generate a *katakana* to English dictionary from raw Japanese text and its English translation, which in many cases is available in electronic form, with a reasonable degree of accuracy. The algorithm thus allows an instructor to generate specialized Japanese vocabularies from selected articles, so that he/she can individualize lessons for a particular student's technical interest and competence at reading *katakana*. The algorithm has been shown to be very effective in technical Japanese instruction and is currently used in a course on Japanese information processing for electrical engineering students at Purdue University.

1. Introduction

Interest in Japanese language instruction has risen dramatically in recent years, particularly for those Americans engaged in technical disciplines. However, the Japanese language is generally regarded as one of the most difficult languages for English-speaking people to learn. While the number of individuals studying Japanese is increasing there remains an extremely high attrition rate, estimated by some to be as high as 80% (Mills *et al.*, 1988). Much of this difficulty can be attributed to the Japanese writing system. Japanese text consists of two distinct orthographies, a phonetic syllabary known as *kana* and a set of logographic characters, originally derived from the Chinese, known as *kanji* (see Fig. 1). The *kana* are divided into two phonetically equivalent but graphically distinct sets, *katakana* and *hiragana*, both consisting of forty-six symbols and two diacritic marks denoting changes in pronunciation. The *katakana* are used primarily for writing words of foreign origin that have been adapted to the Japanese phonetic system. Due to the limited number of *katakana*, their relatively low visual complexity, and their systematic arrangement, memorizing their pronunciations does not represent a significant barrier to the student of Japanese. If the student can also assimilate the phonological transformations that occur, then the effort required to learn *katakana* yields significant returns to readers of technical Japanese due to the high incidence of terms derived from English and transliterated into *katakana* (see Fig. 1).

In this article, we discuss the implementation of an algorithm for automatically generating a *katakana* to English dictionary from raw Japanese text and its English translation. The development of this algorithm was motivated by several factors. First, since the *katakana* orthography is used primarily to write foreign words, most of the newly created *katakana* words are not available in contemporary Japanese to English dictionaries, especially when dealing with specialized technical vocabulary.[1] Also, while these vocabularies are crucial for understanding technical Japanese, they are commonly ignored in conventional Japanese language instruction at universities (Davis and Smith, 1994). An automated method of generating a *katakana* to English dictionary could mitigate this deficiency by allowing an instructor to easily generate specialized Japanese vocabulary lists from selected technical articles. Consequently, the instructor can easily tailor his/her instruction according to each student's particular technical interest. In addition, the entries of these *katakana* to English dictionaries can be used for the development of a student model (Maciejewski and Kang, 1994) which is used by a Japanese language intelligent tutoring system (Maciejewski and Leung, 1992) designed to assist engineers at acquiring proficiency in reading technical Japanese.

The algorithms developed to generate these *katakana* to English dictionaries are illustrated using data that consists of an electronic document of 3000 Japanese phrases and their English translations, obtained through the courtesy of Nippon Telephone and Telegraph (NTT) of Japan (Kang and Maciejewski, 1992). Among the 3000 Japanese phrases in the document, nearly one half contain at least one *katakana* word. For each *katakana* word, the corresponding English phrase is scanned in order to search for the English word from which the *katakana* word was derived. An approximate string matching algorithm (Wagner and Fischer, 1974) is used for finding the English origin of a *katakana* word from the corresponding English text. In applying this algorithm, we discuss two fundamentally different approaches as illustrated in Fig. 2. The one approach is to directly compare the spelling of a transliterated *katakana* word and its English origin. It will be shown, however, that a higher degree of accuracy can be obtained by first converting both the *katakana* and the English into their phonemic representations before applying the string comparison algorithm. In both approaches, one must apply a set of phonological transformation rules to

Correspondence: A. A. Maciejewski, School of Electrical and Computer Engineering, Purdue University, 1285 Electrical Engineering Building, West Lafayette, IN 47907-1285, USA. E-mail: maciejew@ecn.purdue.edu

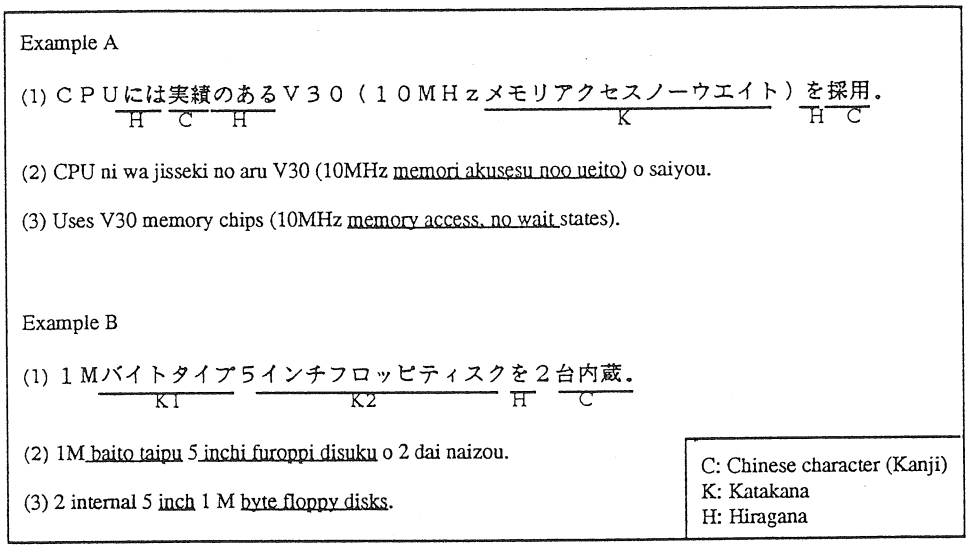


Fig. 1 A sample of Japanese text obtained from the user manual for an NEC personal computer. The Japanese text in the first line of both examples consists of four different orthographies, namely *kanji, kana* (which consists of *katakana* and *hiragana*), and roman characters. The second line in both examples is an approximate pronunciation of the Japanese text, transliterated into roman characters. The third line is the corresponding English translation. Note the frequency of occurrence of the *katakana* characters and the phonetic modifications which occur by comparing the underlined portions of the second and third lines.

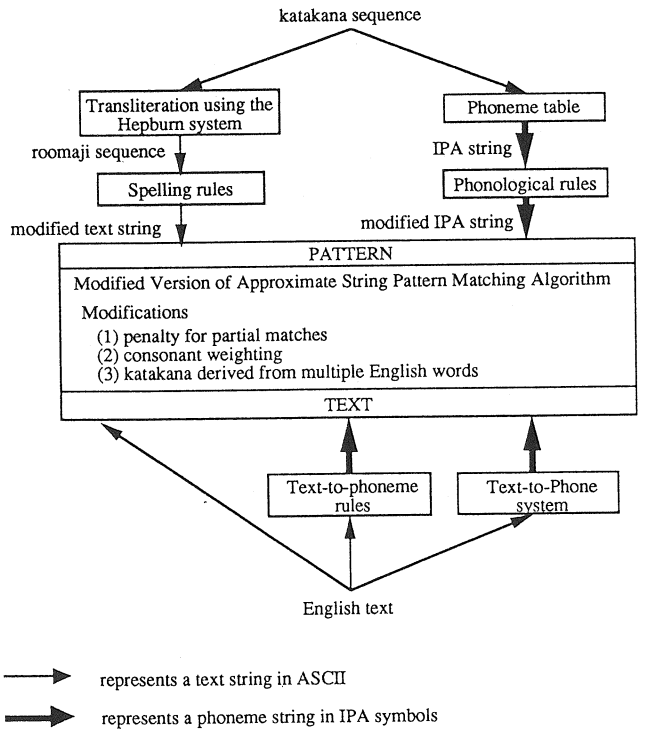


Fig. 2 Overview of the pattern matching process using both a text string and phoneme string. In all cases the 'pattern' to be matched is obtained from the *katakana* in the Japanese phrase and the 'text' is generated from the complete English translation of the Japanese phrase. The most fundamental difference between the different versions of the algorithms implemented is whether the actual pattern matching is performed on ASCII strings or on IPA phoneme strings. In both cases the *katakana* sequence is modified to account for the phonological differences between Japanese and English. The approximate string pattern matching algorithm is also modified to account for linguistic factors that are not phonological in origin.

compensate for the structural differences in phonology between Japanese and English. The pattern matching algorithm is also modified in order to account for a number of factors that are unrelated to the phonological transformation.

The remainder of this article is organized as follows. The next section provides a brief introduction to the *katakana* writing system and to the rules of Japanese phonology. In Section 3 the text pattern matching algorithm is presented. It is followed by a method of phonemic modification for a *katakana* sequence and its English translation. Section 5 describes the phoneme pattern matching algorithm. The performance of the algorithms with various techniques is analysed in Section 6. Finally, the conclusions of this article are provided in the last section.

2. Katakana and Japanese Phonology

The Japanese lexicon contains an extremely large number of words originating from foreign languages. While the proportion of words of Chinese origin in the lexicon is extremely large due to the profound cultural influence of China, words of English origin have dominated the class of loan words since the late 19th century. In a study of Japanese publications performed between 1956 and 1964, over 80% of the foreign words originated from English (Kokuritsu Kikugo Kenkyuujo, 1964). This process of adopting English words into the lexicon is particularly common for relatively new or specialized terms arising in technical literature.

When adopting a word of foreign origin into Japanese, the original pronunciation of that word is typically transliterated into *katakana* which graphically represents all of the possible phonetic sequences in the

Table 1 Katana characters and their phonetic representations in the IPA symbols: Basic syllables

ア	イ	ウ	エ	オ
/a/	/i/	/u/	/e/	/o/
カ	キ	ク	ケ	コ
/ka/	/ki/	/ku/	/ke/	/ko/
サ	シ	ス	セ	ソ
/sa/	/ʃi/	/su/	/se/	/so/
タ	チ	ツ	テ	ト
/ta/	/ʧi/	/tsu/	/te/	/to/
ナ	ニ	ヌ	ネ	ノ
/na/	/ni/	/nu/	/ne/	/no/
ハ	ヒ	フ	ヘ	ホ
/ha/	/hi/	/fu/	/he/	/ho/
マ	ミ	ム	メ	モ
/ma/	/mi/	/mu/	/me/	/mo/
ヤ		ユ		ヨ
/ja/		/ju/		/jo/
ラ	リ	ル	レ	ロ
/ra/	/ri/	/ru/	/re/	/ro/
ワ				
/wa/				

Table 2 Katana characters and their phonetic representations in the IPA symbols: Additional variations of the basic syllables

(a) Modified Syllables

ガ	ギ	グ	ゲ	ゴ
/ga/	/gi/	/gu/	/ge/	/go/
ザ	ジ	ズ	ゼ	ゾ
/za/	/ʤi/	/zu/	/ze/	/zo/
ダ	ヂ	ヅ	デ	ド
/da/	/ʤi/	/zu/	/de/	/do/
バ	ビ	ブ	ベ	ボ
/ba/	/bi/	/bu/	/be/	/bo/
パ	ピ	プ	ペ	ポ
/pa/	/pi/	/pu/	/pe/	/po/

(b) Consonants Plus /ja/, /ju/, or /jo/

キャ	キュ	キョ	ギャ	ギュ	ギョ
/kja/	/kju/	/kjo/	/gja/	/gju/	/gjo/
シャ	シュ	ショ	ジャ	ジュ	ジョ
/ʃa/	/ʃu/	/ʃo/	/ʤa/	/ʤu/	/ʤo/
チャ	チュ	チョ			
/ʧa/	/ʧu/	/ʧo/			
ニャ	ニュ	ニョ			
/nja/	/nju/	/njo/	ビャ	ビュ	ビョ
ヒャ	ヒュ	ヒョ	/bja/	/bju/	/bjo/
/hja/	/hju/	/hjo/	ピャ	ピュ	ピョ
ミャ	ミュ	ミョ	/pja/	/pju/	/pjo/
/mja/	/mju/	/mjo/			
リャ	リュ	リョ			
/rja/	/rju/	/rjo/			

(c) Mora Consonants

ン	ッ
/N/	/Q/

Japanese language. It is this process of modifying English phonetic sequences to conform to the rules of Japanese phonology which presents English-speaking readers of *katakana* with difficulty in identifying a word's meaning. This is due to the fact that the rules of Japanese phonology are quite different from those of English. In particular, Japanese has only five single vowel sounds /aiueo/ in contrast to the large number of vowel sounds in English. These vowels, when combined with the nine Japanese consonants /kstnhmjrw/ constitute forty-four of the forty-six basic sounds in Japanese which are traditionally organized as shown in Table 1. The pronunciation of each *katakana* character in the table is represented using the international phonetic alphabet (IPA) symbols. Additional variations of these basic syllables are presented in Table 2.

The above description of the *katakana* orthography illustrates certain inherent limitations that are imposed on phonetic sequences in the Japanese language. In particular, Japanese does not allow any consonant clusters (except when a consonant is followed by a glide or preceded by a moraic consonant; Vance, 1987). In addition, consonants may not appear at the end of a sequence. These restrictions, together with the limited number of Japanese vowel sounds, result in the vast majority of phonological modifications which occur when transliterating an English word into *katakana*. By the same token, these resulting modifications are the source of difficulty for English-speaking readers of *katakana*.

It should also be noted that there are additional difficulties to comprehending *katakana* unrelated to the phonological processes involved. In particular, while it is true that the vast majority of loan words are created by the phonological process, foreign borrowings may also be modified by changes in form due to simplification, semantics, or Japanese coinage (Shibatani, 1990). For example, simplification frequently occurs with polysyllabic words such as 'television' and 'word processor' which are shortened to the *katakana* words *terebi* and *waapuro*, respectively. Changes in semantics have resulted in the *katakana* word *hochikisu* being used to designate a stapler, whereas its phonetic origin is from 'Hotchkiss', the name of the person who invented the stapler. Examples of coinage which result from combinations of existing loan words include *maikaa* (derived from my + car) and *maihoomu* (derived from my + home) which refer to privately owned cars and houses. However, this work will primarily focus on identifying Japanese scientific terms resulting from phonological modification since the majority of *katakana* belong to this category.

3. Text Pattern Matching

The main process of constructing a *katakana* to English dictionary is searching for the English origin of a *katakana* word from the given English translation of the Japanese text. This is a classic string pattern matching problem if one considers the *katakana* to be the 'pattern' and its English text to be the 'text.' Several algorithms for the string pattern matching problem have been developed by various researchers (Wagner and Fischer, 1974; Boyer and Moore, 1977; Knuth *et al.*, 1977). Among these algorithms, an approximate string matching algorithm (Wagner and Fischer, 1974) is used, since an exact copy of the pattern cannot be expected in this application. This algorithm is based on the dynamic programming technique and was originally designed to find the first *k*-approximate match.

Table 3 The modified Hepburn romanization system

a	i	u	e	o
ka	ki	ku	ke	ko
sa	shi	su	se	so
ta	chi	tsu	te	to
na	ni	nu	ne	no
ha	hi	fu	he	ho
ma	mi	mu	me	mo
ya		yu		yo
ra	ri	ru	re	ro
wa				
ga	gi	gu	ge	go
za	ji	zu	ze	zo
da	ji	zu	de	do
ba	bi	bu	be	bo
pa	pi	pu	pe	po
kya		kyu		kyo
sha		shu		sho
cha		chu		cho
nya		nyu		nyo
hya		hyu		hyo
mya		myu		myo
rya		ryu		ryo
gya		gyu		gyo
ja		ju		jo
bya		byu		byo
pya		pyu		pyo
n̄				

The English transliterations for the *katakana* characters are arranged in the same order as Tables 1 and 2. The mora consonant does not have a unique transliteration but depends on the following consonant.

Table 4 List of spelling transformation rules used after transliteration of a *katakana* word using the Hepburn romanization

spelling transformation rule				example	
				katakana word	English origin
u	→	*	/ C_(C|#)	*shisutemu*	system
o	→	*	/ (d|t)_(C|#)	*doraibaa*	driver
i	→	*	/ C_(C|#)	*matchi*	match
howa	→	wh	/ _V	*howaito*	white
(u|uu)	→	w	/ _V	*uuru*	wool
i	→	y		*iesu*	yes
ee	→	yV		*eeru*	Yale
y	→	*	/ _V	*kyaburetaa*	carburetor
a	→	Vr	/ (oo|o|e|i)_#	*hea*	hair
aa	→	Vr		*misutaa*	mister
a	→	Vr	/ _#	*aakitekucha*	architecture
oo	→	Vr		*pooku*	pork
s	→	c	/ _e	*serori*	celery
s	→	th		*sumisu*	Smith
z	→	j	/ _e	*zerii*	jelly
z	→	th		*mazaa*	mother
j	→	(d|z)		*ejison*	Edison
b	→	v		*banira*	vanilla
h	→	f		*ueha*	wafer
r	→	l		*reñgusu*	length
ts	→	(t|z)		*tsurii*	tree

The rule format $A \rightarrow B/C_1\text{-}C_2$ implies that the string C_1AC_2 may be replaced by C_1BC_2. All lower-case letters in the table represent themselves. The upper-case letters and special characters have the following meanings: C, any consonant; V, any vowel; #, sequence boundary; *, null character; $(x_1 | x_2)$, either x_1 or x_2.

However, for this application one is interested in the best approximate match, so that all entries of the pattern matching table need to be computed (details are in the Appendix).

Before the pattern matching algorithm can be applied the pattern and the text must be in the same orthography so the *katakana* word is transliterated into *roomaji* (roman letters) using the Hepburn romanization system listed in Table 3. The *roomaji* word is then transformed into a string sequence using the set of spelling rules listed in Table 4, in order to more closely match English spelling conventions. The resulting sequence can now be used as a pattern and placed in a column of a pattern matching table with the corresponding English text being put into a row.

When the modified transliterated *katakana* is used in the pattern matching algorithm, the success rate in determining the correct English equivalent was less than 70% on the 1500 phrases tested. The main reason for this relatively poor performance is that the Japanese have adapted English words based primarily on a native speaker's pronunciation and not on a word's spelling. The following sections address this issue by performing the pattern matching on a phonemic version of the Japanese and English texts.

4. Text-to-Phoneme Conversion

In Japanese the *katakana* characters are essentially a phonetic alphabet, so that each symbol has a unique pronunciation. This means that there is a one-to-one correspondence between the Japanese phonetic symbols and the IPA symbols. Therefore it is very simple to get the correct phonemic sequence for *katakana* by using Tables 1 and 2.

On the other hand, converting an English text into its phonetic equivalent presents a problem. An English word can be translated into its phonetic equivalent by various methods: applying a set of phonological rules, using a text-to-speech system such as the MITalk system (Allen *et al.*, 1987), or using a phoneme dictionary. While the simplest and the most accurate method is to use a phoneme dictionary, one that includes technical terminology is not always available.

Another possibility is to use text-to-phoneme rules to get the phoneme sequence for an English word. Assuming that there exist N_i matched rules for the ith string of an English word, there will exist $\Pi_{i=1}^{k} N_i$ possible pronunciations of an English word, which may cause a serious computational burden. However, since an exact phoneme sequence is not necessary for finding the English origin of a *katakana* word, only the most frequently occurring rules need to be applied. When each entry of the pattern matching table is being computed, the English letters in the entry are converted into IPA phonemes. This is done by using a relatively small set of text-to-phoneme rules listed in Table 5a which transform the English alphabet into the most likely IPA phoneme, and the rules listed in Table 5b which represent relatively frequently occurring transformations. The rule base is scanned until all matched rules are found. If any of the matched rules converts an English letter into the same phoneme as the phoneme

Table 5 List of rules used for English spelling to phoneme conversion

(a) English letters and their most likely phonemic representation

English letter	IPA	English letter	IPA
a	/a/	n	/n/
b	/b/	o	/o/
c	/k/	p	/p/
d	/d/	q	/k/
e	/e/	r	/r/
f	/f/	s	/s/
g	/g/	t	/t/
h	/h/	u	/u/
i	/i/	v	/v/
j	/ʤ/	w	/w/
k	/k/	x	/ks/
l	/l/	y	/i/
m	/m/	z	/z/

(b) text-to-phoneme rules

no	text-to-phoneme rule	example	
		English word	phoneme sequence
1	ch → (k | ʃ | ʧ)	chemical	/kemikəl/
		Chicago	/ʃikaːgou/
		bench	/benʧ/
2	sh → ʃ	show	/ʃou/
3	tion → (ʃon | ʧon)	motion	/mouʃən
		question	/kweswʧən/
4	e → * / C_#	white	/hwait/
5	y → j / C_	yes	/jes/
6	ck → k	pick	/pik/
7	c → s	circle	/səːrkl/
8	th → (ð | θ)	father	/faːðər/
		length	/leŋθ/
9	gh → *	neighbor	/neibər/
10	s → z	easy	/iːzi/
11	g → ʤ	energy	/enərʤi/

The rule format $A \rightarrow B/C_1\text{-}C_2$ implies that the string C_1AC_2 may be replaced by C_1BC_2. All lower-case letters in the table represent themselves. The upper-case letters and special characters have the following meanings: C, any consonant; V, any vowel; #, sequence boundary; *, null character; $(x_1 \mid x_2)$, either x_1 or x_2.

in a *katakana* sequence corresponding to the entry being computed, then both the phonemes corresponding to the entry are considered as being matched. Thus, the pattern matching algorithm is modified as shown in Fig. 3a.

Despite the fact that the accuracy of the English text to phoneme conversion is relatively poor, this phoneme pattern match still performs much better than the straight text pattern match. In order to improve the English text to phoneme conversion process, the text-to-phone system (Elovitz *et al.*, 1976), developed by the Naval Research Laboratory (NRL) using the 50,000-word standard corpus of present-day edited American English (Brown Corpus) (Kucera and Francis, 1967), was also applied to translate an English word into its phonetic equivalent. This system consists of a set of 329 letter-sound rules that translate English text into its IPA equivalent. The rules produce correct pronunciations for approximately 90% of the words in an average sample of English text. Although the system is not the best of the text-to-speech systems that have been developed, the performance of the system is good enough to translate an English word into a phoneme sequence and match the sequence with the phoneme sequence of a *katakana* word. While the text-to-phone system greatly improves the accuracy of English text to phoneme conversion, it will be shown that the accuracy of the *katakana* matching process is not particularly sensitive to small numbers of errors in phoneme conversion.

5. Phoneme Pattern Matching

5.1 Phonological Rules

Using direct pattern matching between phoneme strings results in a relatively low success rate at match-

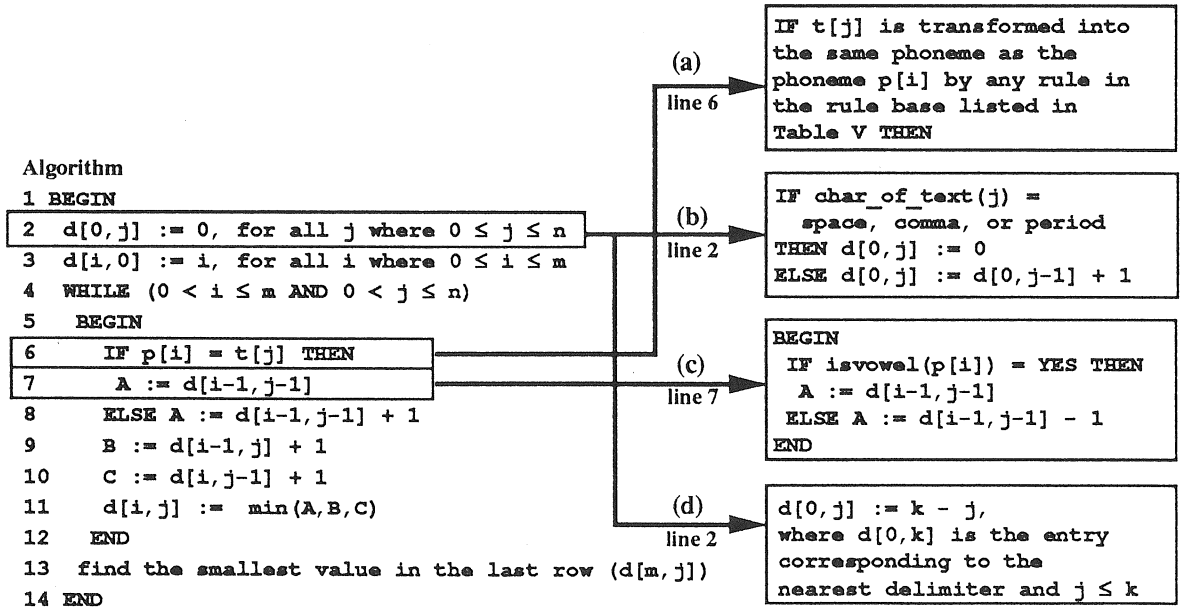


Fig. 3 The pseudocode implementation of the modified approximate pattern matching algorithm. Modification (a) is to account for the fact that the Japanese have adapted English words based primarily on a native speakers pronunciation. Modification (b) is to allow for partial matches due to simplification. Modification (c) is to account for the higher correlation in consonants between Japanese and English. Modification (d) is to determine where the beginning of the matching pattern is located. (Modification (d) is applied after the pattern and text have been reversed).

Table 6 List of phonological transformation rules used on the phonemic representations of a *katakana* sequence to approximate the phonemic representation of an English word

no	phonological rule				example: katakana word	example: English origin
1	u	→	*	/ C_(C|#)	/ʃisutemu/	/sistəm/
2	o	→	*	/ (d|t)_(C|#)	/doraibaː/	/daivər/
3	i	→	*	/ C_(C|#)	/maQʧi/	/mæʧ/
4	howa	→	hw	/ _V	/howaito/	/hwait/
5	(u|uː)	→	w	/ _V	/uːru/	/wul/
6	i	→	j		/iesu/	/jes/
7	eː	→	jV		/eːru/	/jeil/
8	j	→	*	/ _V	/kyaputeN/	/kæptin/
9	a	→	Vr	/ (oː|o|e|i)_#	/hea/	/heər/
10	aː	→	Vr		/misutaː/	/mistər/
11	a	→	Vr	/ _#	/aːkitekuʧa/	/arkiteʧər/
12	oː	→	Vr		/poːku/	/pərk/
13	s	→	θ		/sumisu/	/smiθ/
14	z	→	ʤ	/ _e	/zeriː/	/ʤeli/
15	z	→	ð		/mazaː/	/maðər/
16	ʤ	→	(d|z)		/eʤisoN/	/edəsən/
17	b	→	v		/banira/	/vənilə/
18	h	→	f		/ueha/	/weifər/
19	r	→	l		/reNgusu/	/leŋθ/
20	ts	→	(t|z)		/tsuːri/	/triː/
21	ʃ	→	s		/ʃisutemu/	/sistəm/
22	ʧ	→	t		/maruʧi/	/məlti/
23	oru	→	Vr	/ C_	/ʧuːdoru/	/tudər/
24	eru	→	Vr	/ C_	/enerugiː/	/enərʤi/

The rule format $A \rightarrow B/C_1\text{-}C_2$ implies that the string C_1AC_2 may be replaced by C_1BC_2. All lower-case letters in the table represent themselves. The upper-case letters and special characters have the following meanings: C, any consonant; V, any vowel; #, sequence boundary; *, null character; $(x_1 | x_2)$, either x_1 or x_2.

ing a *katakana* sequence with its English origin because of the phonological differences between Japanese and English. The set of phonological rules presented in Table 6 is used in order to help improve the phoneme pattern matching. The phonological rules, which generally correspond to the spelling rules listed in Table 4, represent the most frequently occurring transformation rules that an English word undergoes when adopted into the Japanese lexicon. A *katakana* IPA string is transformed by these phonological rules in order to approximate the differences between Japanese and English phonology. Since more than one rule can be matched with each phoneme, a large number of modified strings may be generated in some cases. All resulting phoneme strings are compared with an English phoneme string obtained by using the text-to-phone system or the text-to-phoneme rules. This phoneme pattern matching scheme performs significantly better than the straight text pattern matching scheme. Using these rules the success rate of the algorithm went up to 80%. Although there are still quite a few errors in the transformation of the English text into a phoneme sequence, these errors do not account for the 20% of *katakana* that are not correctly matched. These remaining mismatches are due to a number of factors that are unrelated to the phonological transformation. The following sections address how to improve this success rate.

5.2 Penalty for Partial Match

One of the error sources is due to partial matches, which occur when a match starts in the middle of a word. Since a match to the end of an English word is likely to be an error, the algorithm is modified to put more emphasis on matching at the beginning of a word which is motivated by the process of simplification mentioned previously. This is done by modifying the initialization of the pattern matching table in the following manner: if the character in the text is a delimiter such as a space, coma, or period, the entry in the zeroth row of the column is set to zero; otherwise it is set to the distance from the initial character of the word. This change can be implemented by modifying the algorithm as shown in Fig. 3b. An example of how the performance of the pattern matching algorithm has been improved is illustrated in Fig. 4.

Pattern

		d	i	s	t	ə	n	t		n	ʌ	l		c	ə	m	p	o	u	z
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1
l	2	2	2	2	2	2	1	1	2	1	1	1	2	2	2	2	2	2	2	2

Text

(a) Results of applying the original pattern matching algorithm.

		d	i	s	t	ə	n	t		n	ʌ	l		c	ə	m	p	o	u	z
	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7
n	1	1	2	3	4	5	5	6	1	0	1	2	1	1	2	3	4	5	6	7
l	2	2	2	3	4	5	6	6	2	1	1	1	2	2	2	3	4	5	6	7

(b) Results of applying the modified pattern matching algorithm with a penalty for partial matches

Fig. 4 An example of the modified pattern matching algorithm incorporating a penalty for partial matches. In this example, the *katakana* word *nuru* is compared with the English text 'distant null compose . . . .' After applying the phonological rules of Table 6, there exist eight possible approximations for *nuru* which are nul, nur, nulu, nuru, nl, nlu, nru, and nr. Among these approximations, the smallest number of mismatches is one character. The above two tables illustrate the case where 'nl' is matched with the English text. In (a) the *katakana* word is matched with the English word 'distant' and 'null' with the same number of mismatches; however, when the penalty for partial matches is introduced in (b) only the correct translation 'null' is matched with the *katakana* word.

5.3 Consonant Weighting

As mentioned previously, Japanese has a much smaller set of vowels as compared to the large number of vowels in English. While the number of Japanese consonants is also smaller than the number of English consonants, there exists a much higher correlation between Japanese and English consonants. Hence, a match of consonants between Japanese and English is more likely to occur than a match of vowels for the two languages. The pattern matching algorithm is therefore changed as shown in Fig. 3c to put more emphasis on a match of consonants.

5.4 Katakana from Multiple English Words

A problem also occurs when a *katakana* sequence originated from multiple English words rather than a single word due to the lack of word-boundaries in Japanese writing. A common example of this is the katakana *waapuro* which is derived from 'word processor'. Since the pattern matching algorithm is originally designed to point to the end of the matching pattern, it also needs to find out where the beginning of the matching pattern is located. Therefore, a simple way to do this is to run the pattern matching algorithm again on the reversed sequences of the pattern and text. The same penalty for a partial match must still be included in order to make sure that the match begins at the point identified as a sequence boundary. This is done by initializing the pattern matching table to the distance from the last character of the word as shown in Fig. 3d.

6. Results

This section will analyse the performance of the algorithms described and discuss why certain errors occurred and how the modifications address these errors. The accuracy of each algorithm is measured in terms of the number of errors where an error is defined as the number of times a *katakana* word is not matched with its phonological origin word in the English translation text. The case where the phonological origin words do not appear because of a change in semantics is not counted as an error. This occurred in approximately 3% of the entries in the NTT source.

Various techniques were tried to improve the performance of the pattern matching algorithm as summarized in Fig. 2. First, the plain text pattern match was tried by comparing a roomaji word transliterated from a *katakana* sequence with the corresponding English text using the pattern matching algorithm presented in the Appendix. Since there is little correspondence between the English spelling of a word and its transliterated Japanese, the success rate of this technique is extremely low, i.e. about 65%. In order to transform the *katakana* string into a string more closely resembling the spelling of the English text, a simple set of rules to approximate changes in phonetics represented in roman characters was used. Using these rules the success rate improved, but only to 70%. The main reason for this relatively poor performance is that the Japanese have adapted English words based primarily on a native speaker's pronunciation and not on a word's spelling. Hence, phoneme strings of both a *katakana* sequence and its English text were tried to be matched. Since there exist differences in the phonological structure between Japanese and English that cause a relatively large number of errors, straight phoneme pattern matching did not improve the performance of the algorithm. In order to compensate for these differences between the two languages, the set of phonological transformation rules listed in Table 6 were applied to the *katakana* IPA string. This phoneme pattern matching greatly improved the success rate as compared with the unmodified phoneme pattern matching, bringing the efficiency up to 80%.

The remaining 20% of the cases that caused errors consist of mismatches due to a number of factors that are unrelated to the phonological transformation. One of these factors is due to the lack of word-boundaries in Japanese writing which creates *katakana* sequences that originate from multiple English words rather than a single word. This occurs in approximately 10% of the entries and was addressed by re-applying the algorithm to the reversed input strings. A problem also occurs when *katakana* words are created by the simplification process and then quite often coined with other existing *katakana* words. The *katakana* words in this category account for about 5% of the 1500 entries in the NTT source. These errors are eliminated by incorporating the penalty for a partial match and by putting more emphasis on the match of consonants. Table 7 compares the total number of errors with and without each of these two techniques for the 1500 trials. In comparing an extremely simple set of text-to-phoneme rules used for transforming an English word into a phonetic representation the performance of the algorithm using these rules is not that much worse than the algorithm using a much more complex text-to-phone system. The reason for this result is due to the fact that one is looking for the best approximate match rather than an exact match, hence, an English word that is not correctly transformed into phonemes does not cause the algorithm to perform poorly if only a few phonemes are incorrect. This is especially true when a consonant weighting technique is used because consonants are typically transformed properly in the text-to-phoneme rules and the text-to-phone system. Consequently, the number of errors is dramatically reduced since the consonants are more likely to be correctly matched than the vowels. In addition, incorporating the penalty for a partial match reduces the number of errors by eliminating the possibility of starting a match in the middle of a word.

The best matching accuracy occurred when the text-to-phone system was used for an English phoneme and all modifications were applied to the pattern matching algorithm with the three techniques described. In this

Table 7 Result of phoneme pattern matching algorithm

English phoneme string obtained from	penalty on partial match	weight on consonant	no of errors
text-to-phoneme* rules	off	off	26
	off	on	14
	on	off	12
	on	on	2
text-to-phone† system	off	off	11
	off	on	9
	on	off	4
	on	on	1

All possible phonological approximations of each *katakana* sequence are matched with the phonetic equivalents of the English text obtained using both the text-to-phone system and the text-to-phoneme rules listed in Table 5. The total number of errors not being matched with correct English translation of the *katakana* sequence from 1500 trials is presented with a penalty on a partial matching and consonant weighting on and off.
*Using the rules specified in Table 5
†The text-to-phoneme system developed by the Naval Research Laboratory.

case, only one mismatch occurs which is when the *katakana* word *roiko* is matched with the English text 'leuco type thermosensitive recording paper.' While its correct phonological origin is 'leuco', the English word 'recording' is matched with *roiko* with only one phoneme mismatch. Case 1 below presents the best possible match between leuco and *roiko*, where there are two mismatches. On the other hand, in Case 2 there is only one mismatch between recording and *roiko*.

Case 1

leuco →text-to-phone→ /ljuko/
roiko →Tables 1, 2→ /roiko/ →Table 6 rule 19→ /loiko/

Case 2

recording →text-to-phone→ /rikordIŋ/
roiko →Tables 1, 2→ /roiko/ →no rule→ /roiko/

This is one case where the modification to accommodate partial matches actually hurts the performance of the algorithm. If one forces a *katakana* sequence to match the entire English word, this error would not occur, however, there is no way to know when or where a word is abbreviated. This example serves to illustrate the inherent difficulty in ever achieving perfect performance without including some form of semantic processing.

In order to examine the tradeoffs between matching accuracy and computational expense, tests were performed where not all of the phonological approximations of each *katakana* sequence are used. By ordering the phonological rules based on their probabilities, the most probable patterns can be obtained. Table 8 shows the results of this study illustrating the expected tradeoff, i.e. using more phonological rules increasing the accuracy of the result as well as the computation time required.

Table 8 Tradeoffs between matching accuracy and computational expense

patterns resulting from applying all rules	execution time (sec) / no. of errors			
	English text transformed using			
	text-to-phoneme rules		text-to-phone system	
only the most likely pattern	226.53	18	258.10	8
only the top 5% most likely patterns	249.67	16	266.78	8
only the top 25% most likely patterns	356.25	8	389.61	3
only the top 50% most likely patterns	472.84	3	509.48	1
all patterns	537.97	2	592.13	1

The different number of phonological approximations (patterns) of each *katakana* word that are matched with an English phoneme are tested.

7. Conclusions

The goal of this work was to design an algorithm for automatically generating a *katakana* to English dictionary from raw Japanese text and its English translation. The implementation and validation of the different versions of this algorithm has shown that matching the phoneme versions of the *katakana* string and its English origin produces much better results as compared with directly matching transliterated *katakana* with the English text. In both cases, the set of phonological rules used to compensate for phonological differences between Japanese and English was extremely important. It was also illustrated that the approximate pattern matching algorithm was greatly improved by using various techniques such as a penalty for partial matches, consonants weighting, and finding *katakana* from multiple English words.

The development of this algorithm has been shown to be very effective for technical Japanese instruction by allowing an instructor to generate specialized Japanese vocabulary lists from selected articles. This facilitates individualized instruction by matching a student's technical interest and reading competence in *katakana*

Acknowledgements

The authors would like to thank the anonymous reviewers for their many useful suggestions which have improved the final version of this article. This material is based upon work supported by the National Science Foundation under Grant No. INT-8818039 and in part by the NEC Corporation and a Purdue University Global Initiative Grant.

Notes

1. We know of two notable exceptions. The first is a multiple volume set of technical dictionaries covering several disciplines that was compiled by the Japanese Ministry of Education, Science, and Culture (Monbushō). It is manufactured in a CD-ROM format by the National Center for Science Information Systems. The second is the Electronic Dictionary project entitled 'Research on Electronic Dictionaries for Natural Language Processing' that is being conducted by the Japan Electronic Dictionary Research Institute, Ltd. (EDR) which was established in 1986 and sponsored by the Japan Key Technology Center and eight private corporations.

References

Allen, J., Hunnicutt, M. S. and Klatt, D. (1987). *From Text to Speech: The MITalk System*. Cambridge University Press, New York.

Boyer, R. S. and Moore, J. S. (1977). A Fast String Searching Algorithm. *Communications of ACM*, 20: 762–72.

Davis, J. L. and Smith, T. W. (1994). Computer-Assisted Distance Learning, Part I: Audiographic Teleconferencing, Interactive Satellite Broadcasts, and Technical Japanese Instruction from the University of Wisconsin–Madison. *IEEE Transactions on Education*, E-37(2): 228–33.

Elovitz, H., Johnson, R., McHugh, A. and Shore, J. (1976). Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, ASSP-24: 446–59.

Kang, Y.-S. and Maciejewski, A. A. (1992). Data on English to Japanese Transliteration of Technical Terminology.

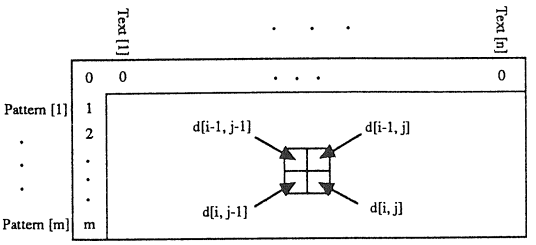
Technical Report TR-EE 92-34, Purdue University, West Lafayette, IN.
Knuth, D. E., Morris, J. H. and Pratt, V. R. (1977). Fast Pattern Matching in Strings. *SIAM Journal on Computing*, 6: 323–50.
Kokuritsu Kokugo Kenkyuujo (1964). *Gendai-zasshi 90shu no yoogo yooji (3)*. Number 25.
Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English.* Brown University Press, Providence, RI.
Maciejewski, A. A. and Kang, Y.-S. (1994). A Student Model of Katakana Reading Proficiency for a Japanese Language Intelligent Tutoring System. IEEE Transactions on Systems, Man, and Cybernetics, SMC-24: 1347–57.
Maciejewski, A. A. and Leung, N. K. (1992). The Nihongo Tutorial System: An Intelligent Tutoring System for Technical Japanese Language Instruction. *Journal of Computer Assisted Language Learning and Instruction Consortium*, 9: 5–25.
Mills, D. O., Samuels, R. J. and Sherwood, S. L. (1988). Technical Japanese for Scientists and Engineers: Curricular Options. Technical Report MITJSTP WP 88-02, MIT, Cambridge, MA.
Shibatani, M. (1990). *The Languages of Japan.* Cambridge University Press, New York.
Vance, J. T. (1987). *An Introduction to Japanese Phonology.* State University of New York Press, Albany.
Wagner, R. A. and Fischer, M. J. (1974). The String-to-String Correction Problem. *Journal of ACM*, 21: 168–173.

Appendix

Approximate String Matching Using Dynamic Programming

Let $P = p[1]\, p[2] \ldots p\,[m]$ represent a pattern and $T = t[1]\, t[2] \ldots t[n]$ represent the text (Wagner and Fischer, 1974). The number of symbols in the text, n, is assumed to be large relative to the number of symbols in the pattern, m. Let k be a nonnegative integer. A k-approximate match is a match of P in T that has at most k differences. The differences may be any of the following three types: (i) the corresponding symbols in P and T are different; (ii) P is missing a symbol that appears in T; (iii) T is missing a symbol that appears in P. The inputs for the problem are P, T, and k.

For the dynamic programming solution, $d[i, j]$ is defined as the minimum number of differences between $p[1] \ldots p[i]$ and a segment of T ending at t[j]. The structure of the pattern matching table is shown in Fig. 5a. There will be a k-approximate match ending at $t[j]$ for any j such that $d[m, j] \leq k$. The rules for computing entries of d consider each of the possible differences that may occur at $p[i]$ and $t[j]$ and, of course, the possibility that those two characters may match. The approximate string matching algorithm is shown in Fig. 3. The example presented in Fig. 5b is to match an English word 'happy' to an English sentence 'Have a hsppy day.' The least number of mismatches is 1 when 'happy' matches 'hsppy'.



(a) The structure and initialization of the pattern matching table.

		H	a	v	e		a		h	s	p	p	y		d	a	y	.
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
a	2	2	1	2	2	2	1	2	1	1	2	2	2	2	2	1	2	2
p	3	3	2	2	3	3	2	2	2	2	1	2	3	3	3	2	2	3
p	4	4	3	3	3	4	3	3	3	3	2	1	2	3	4	3	3	3
y	5	5	4	4	4	4	4	4	4	4	3	2	1	2	3	4	3	4

(b) An example of the completed pattern matching table.

Fig. 5 An illustration of the approximate pattern matching algorithm. (a) shows the structure of the pattern matching table along with its initial state. In (b) the table is shown completed for the case where the pattern 'happy' is matched with the text 'Have a hsppy day.' The optimal match location is indicated by the underlined 1 in the bottom row, indicating that 'happy' matches with 'hsppy' with a single mismatched character.