Proceedings of the IASTED International Conference
ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING
July 24-26, 2000, Banff, Canada

EVALUATION OF NUMBER-KANJI TRANSLATION METHOD USING INDUCTIVE LEARNING ON E-MAIL

MASAFUMI MATSUHARA, KENJI ARAKI and KOJI TOCHINAI
Graduate School of Engineering, Hokkaido University,
Kita 13 Nishi 8, Kita-ku, Sapporo 060-8628 Japan.
Phone & Fax: +81-11-706-7389
E-mail: {matuhara, araki, tochinai}@media.eng.hokudai.ac.jp

ABSTRACT

Opportunities and needs are increasing to input Japanese sentences for e-mail on mobile phones since performance of mobile phones is improving and e-mail has come into wide use recently. We need to input Japanese sentences by only 12 keys on mobile phones. We have proposed a method to input Japanese sentences on mobile phones quickly and easily. We call this method Number-*Kanji* translation method using inductive learning. The number strings inputted by a user are translated into *Kanji-Kana* mixed sentences. Since there are many kinds of fields in e-mail, it is difficult to translate number strings into correct sentences fitting the target field. The system based on this method is able to acquire segments as words and dynamically adapt to the fields by its own learning ability. The rate of the correct translation was about 75[%] on an experiment. The user must proofread the erroneous characters in the translation results for the intended sentences. The proofreading needs a large number of key presses. However, the erroneous characters decrease since the system based on this method is able to dynamically adapt to various fields of e-mail. Thus, the number of key presses decreases. This paper shows the evaluation results for the number of key presses in our proposed method on e-mail.

Keywords: Natural language processing, Inductive learning, Japanese, Number-*Kanji* translation, Mobile phone and E-mail

1 INTRODUCTION

Ordinary Japanese sentences are expressed by two kinds of characters: i.e. *Kana* and *Kanji*. *Kana* is Japanese phonogramic characters and has about fifty kinds. *Kanji* is ideographic Chinese characters and has about several thousand kinds. Therefore, we need to use some *Kanji* input methods in order to input Japanese sentences into computers. A typical method is the *Kana-Kanji* translation method of non-segmented Japanese sentences. This method translates non-segmented *Kana* sentences into *Kanji-Kana* mixed sentences. Since one *Kana* character is generally inputted by combination of a few alphabets, this method needs twenty-six keys for the alphabets.

Recently, performance of mobile computing devices is greatly improving. We consider that the devices are grouped into two by their quality. One gives importance to easy operation, the other gives importance to good mobility. Mobile phones are usable as mobile computers and belong to the latter group. Their mobility is very good because typical size of them is small. However, a general mobile phone has only 12 keys, which are 0, 1, ..., 9, * and #, because of the limited size.

The letter cycling input method is most commonly used for the input of sentences on mobile phones. In this input method, a chosen key represents a consonant and the number of pressing it represents a vowel. For example, the chosen key "7" represents "m" and three presses of the key represent "u". Then, the number of key presses is three for the input character "む (*mu*)". Since this input method needs several key presses per a *Kana* character, it is troublesome for a user. Opportunities and needs are rapidly increasing to input Japanese sentences into a small device such as a mobile phone since performance of mobile phones is improving and e-mail has come into wide use recently. Therefore, methods are demanded which enable us to promptly and easily input Japanese sentences for e-mail on mobile phones.

Kushler previously proposed T9®. T9® enables us to input one alphabet per one key press on

*Tegic Communications Inc. has developed T9®. http://www.tegic.com/

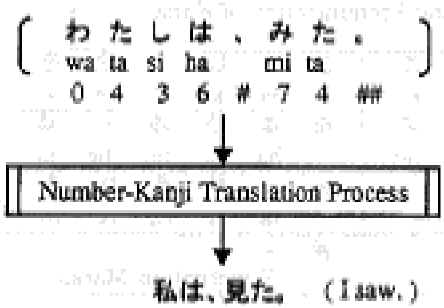

Figure 1: Example of Translation

the keypad of 9 keys[1]. Since three or four letters are assigned to each key of 9 keys, the specific letter intended by one key press is ambiguous. This system disambiguates the pressed keys on word-level. However, it is difficult for Japanese because Japanese sentences are not segmented into words ordinarily. Moreover, the system needs several key presses for input of one *Kana* character because almost all *Kana* characters are expressed by combination of a few alphabets. Higashida has proposed "The degenerated input method"[2]. This input method enables us to input one *Kana* character per one key press because about five *Kana* characters are assigned to each key of 12 keys. In this method, a user is able to input keywords that are "YES", "NO", city names, personal names, and so on. However, non-segmented sentences are not able to be inputted.

We have proposed "*Kana-Kanji* Translation Method Using Inductive Learning"[3]. The system based on this method generates a dictionary adapted to a target field by inductive learning. We consider that this method is effective for a small device such as a mobile phone whose memory is limited generally. Then, we have proposed "Non-Segmented Kana-Kanji Translation Method Using Inductive Learning with Degenerated Keyword Input"[4]. This method enables us to input Japanese sentences promptly and easily. We call this method Number-*Kanji* Translation Method Using Inductive Learning. This method is expressed as IL-NKT in this paper. Figure 1 shows an example of the translation in IL-NKT. A user inputs a string of numbers corresponding to the pronunciation of an intended Japanese sentence by the degenerated input method. The *Kana-Kanji* translation method translates a *Kana* sentence, whereas the number-*Kanji* translation method translates a string of numbers. A key pressed on the keypad of 12 keys represents a line of the 50 sounds table of *Kana*, which is the Japanese syllabary. A user is able to input one

Table 1: 50 Sounds Table of Kana

		k	s	t	n	h	m	y	r	w
a	あ	か	さ	た	な	は	ま	や	ら	わ
i	い	き	し	ち	に	ひ	み		り	
u	う	く	す	つ	ぬ	ふ	む	ゆ	る	
e	え	け	せ	て	ね	へ	め		れ	
o	お	こ	そ	と	の	ほ	も	よ	ろ	を
n										ん

Kana character per one key press by the degenerated input. Table 1 shows the 50 sounds table. It is set in a five-by-ten matrix. The matrix has five vowels and ten consonants. Almost all *Kana* characters are composed of a consonant plus a vowel. Table 2 shows the correspondence of the number with *Kana* characters: e.g. the key "7" represents "ま (*ma*)" or "み (*mi*)" or "む (*mu*)" or "め (*me*)" or "も (*mo*)" of *Kana* characters. The characters in parentheses represent the pronunciation of *Kana*. Then, a number character of 12 keys generally corresponds to a consonant. Since the vowel information degenerates, the string of numbers has ambiguity. The system based on IL-NKT uses inductive learning and information of neighboring characters for the disambiguation. The system is able to acquire segments as words automatically by inductive learning and translate a string of numbers into the *Kanji-Kana* mixed sentence in consideration of connection of the segments by information of neighboring characters. The information of neighboring characters is based on n-gram statistics. Nagao and Mori previously showed a new method of n-gram statistics[5]. Thus, IL-NKT recovers the information lost by the degeneration and translates the strings of numbers into *Kanji-Kana* mixed sentences.

Since a user inputs various messages for e-mail, there are many fields and a target field changes frequently on e-mail. When a target field changes to a new field, sentences of the new field have segments unregistered into the dictionary of the system. The unregistered segments are acquired by the learning ability of IL-NKT. Then, IL-NKT is able to adapt to the target field dynamically. However, the translation result generally has errors. The errors are proofread using the *Kana-Kanji* translation method by the user. The *Kana* characters are inputted using the letter cycling input method for the erroneous characters in the translation result. This input method needs some key presses per one *Kana*. However, the number of key presses decreases because the rate of the correct translation increases and the proofread characters decrease by the adaptability of IL-NKT. This paper outlines IL-NKT and the evaluation experiment for the number of key presses in IL-NKT on e-mail.

Table 2: Correspondence of Number to Kana and Pronunciation of Kana

1:	あ	い	う	え	お	2:	か	き	く	け	こ	3:	さ	し	す	せ	そ
	a	i	u	e	o		ka	ki	ku	ke	ko		sa	si	su	se	so
4:	た	ち	つ	て	と	5:	な	に	ぬ	ね	の	6:	は	ひ	ふ	へ	ほ
	ta	ti	tu	te	to		na	ni	nu	ne	no		ha	hi	hu	he	ho
7:	ま	み	む	め	も	8:	や	ゆ	よ			9:	ら	り	る	れ	ろ
	ma	mi	mu	me	mo		ya	yu	yo				ra	ri	ru	re	ro
*:	゛	゜				0:	わ	を	ん		ー	#:	、	。			
	Voiced Sound, P-Sound						wa	wo	n				Punctuation Marks				

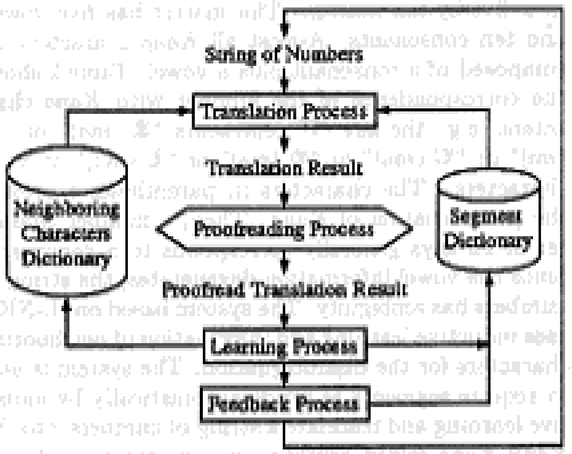

Figure 2: Procedure

2 OUTLINE OF IL-NKT

Figure 2 shows the procedure for IL-NKT. The procedure consists of the translation process, the proofreading process, the learning process and the feedback process in this order.

A user inputs a string of numbers corresponding to the pronunciation of an intended Japanese sentence by only 12 keys. The user is able to input one *Kana* character per one key press by the degenerated input. Table 3 shows an example of input. The Japanese sentence intended by the user is "私は野球を楽しむ (I enjoy baseball.)" in Table 3. The characters in parentheses represent the English sentence for the Japanese sentence. The *Kana* sentence corresponding to the Japanese sentence is "わたしはやきゅうをたのしむ [watasihayakiyuuwotanosimu]". The characters in brackets represent the pronunciation of *Kana*. The string of numbers corresponding to the *Kana* sentence is "0436828104537". The string of numbers needs only 13 key presses for the input. In the translation process, the inputted string of numbers is translated into a *Kanji-Kana* mixed sentence by using the segment dictionary. The segments in the segment dictionary are acquired in the learning process. They are classified into five ranks which are MS, CS, S1, RS and LS. MS is the most certain segment. CS is the common segment. S1 is the segment one. RS is the remained segment. LS is the least certain segment. The order of higher credibility is MS, CS, S1, RS and LS. The segments are applied in this order. Their credibility is evaluated by the credibility evaluation function if there are some candidates of the segment in the same rank. The credibility evaluation function is expressed as CEF and defined as:

$$CEF = \alpha \times ND + \beta \times CR - \gamma \times ER \quad (1)$$

where α, β and γ are coefficients. CR is the rate of the correct translation. ER is the rate of the erroneous translation. ND is the appearance degree for the character strings neighboring the segment. CR and ER are based on the segment dictionary and are updated in the feedback process. ND is based on the neighboring characters dictionary. The credibility for the segment is higher when ND is higher, CR is higher and ER is lower. If the translation result has errors, the proofreading process is performed. The user judges whether it is correct or not and proofreads them. In the learning process, segments are extracted by comparing the input string and its proofread translation result. Table 4 shows the example of the extraction of segments. In Table 4, the number-*Kanji* mixed sentence is the sentence replaced *kana* characters in the *Kanji-Kana* mixed sentence to numbers. They are compared using their common segments. Their common segments are the underline parts in Table 4. The remained segments are between the common segments. The common and remained segments are registered into the segment dictionary as words in IL-NKT. The system extracts a common segment and a remained segment between two segments in the segment dictionary again. When one segment is included in another segment, one segment is CS and the other segment excluded CS is RS. In Table 5, one segment is (8281:野球) and another

Table 3: Example of Input

Intended Japanese Sentence English	私は野球を楽しむ I enjoy baseball.												
Kana Sentence	わ	た	し	は	や	き	ゅ	う	を	た	の	し	む
Pronunciation	wa	ta	si	ha	ya	ki	yu	u	wo	ta	no	si	mu
Input String	0	4	3	6	8	2	8	1	0	4	5	3	7

Table 4: Extraction Example of Segments

Input String	
String of Numbers	0436828104537
Its Proofread Result	
Kanji-Kana Mixed	私は野球を楽しむ
Number-Kanji Mixed	私6野球0楽37
English	I enjoy baseball.
Extraction Result	
Segments	English
(043:私)	I
(6:は)	Postposition
(8281:野球)	Baseball
(0:を)	Postposition
(45:楽)	Enjoy
(37:しむ)	Inflectional ending

segment is (82813*81:野球場). (8281:野球) is included in (82813*81:野球場). The underline part represents the common segment in Table 5. Then, (8281:野球) is extracted as CS and (3*81:場) is extracted as RS. (82813*81:野球場) is deleted. At the same time, all the character strings in the input string and its proofread result are registered into the neighboring characters dictionary. ND is calculated based on the neighboring characters dictionary and is defined as:

$$\text{Segment String} \quad \cdots a_{x-1} \cdot a_x \cdot a_{x+1} \cdots$$

$$ND(a_x) = len(a_{x-1}) \times P_{l(a_x)}(a_{x-1}) + len(a_{x+1}) \times P_{r(a_x)}(a_{x+1}) \quad (2)$$

where $len(X)$ is the length of a segment X, $P_{r(X)}(Y)$ and $P_{l(X)}(Y)$ are the value for a segment Y in the probability distribution on the right side of X and on the left side of X. In the feedback process, the certainty degree for the segment in the segment dictionary is updated. When a translated segment is correct, its CR increases because its certainty degree increases. When a translated segment is erroneous, its ER increases because its certainty degree decreases. It is judged by comparing the translation result and its proofread result. Thus, the system based on this method is improving by the repetition of these processes.

Table 5: Example of CS and RS

S1's	
S1 English	(82813*81:野球場) A baseball ground
S1 English	(8281:野球) Baseball
CS	
CS English	(8281:野球) Baseball
RS	
RS English	(3*81:場) A ground

3 PROOFREADING PROCESS IN IL-NKT

The user proofreads the erroneous characters in the translation result. This process consists of the judgment of the translation result, the input of *Kana* and the *Kana-Kanji* translation.

3.1 JUDGMENT OF TRANSLATION RESULT

The user judges whether the translation result is correct or not. When the translation result has errors, the user chooses the erroneous characters. In Table 6, the underline part is erroneous. Then, the user chooses the string "禁止 0" in the translation result.

3.2 INPUT OF KANA

The *Kana* characters are inputted for the proofreading of the characters chosen by the user. The input of *Kana* is performed by the letter cycling input method on the keypad of 12 keys. The letter cycling input method is most commonly used for input of sentences on mobile phones and needs some key presses per a

Table 6: Example of Proofreading

Input String	
String of Numbers	043682810203039
Intended Sentence	
Kanji-Kana Mixed	私は野球を観戦する
English	I watch baseball.
Translation Result	
Kanji-Kana Mixed	私は野球を <u>禁止 0</u> する
Input of Kana	
Kana	か ん せ ん
Pressed Keys	2 000 3333 000

Kana character. In Table 6, the *Kana* string is "かんせん [kansen]" for the proofreading of "禁止 0". The number of key presses is 11 for the four *Kana* characters.

3.3 KANA-KANJI TRANSLATION

The *Kana-Kanji* translation method translates *Kana* characters into the correct words by using the proofreading dictionary. The proofreading dictionary is different from the segment dictionary. The segment dictionary is used for the number-*Kanji* translation, whereas the proofreading dictionary is used for the *Kana-Kanji* translation. Since a *kana* character is less ambiguous than a number character, the number of the word candidates on the *Kana-Kanji* translation is less than that on the number-*Kanji* translation. Therefore, the number of key presses for the *Kana-Kanji* translation is less than that for the number-*Kanji* translation since the user chooses a correct word in the word candidates. For example, a number string "2030" is translated by the number-*Kanji* translation method and a *Kana* string "かんせん [kansen]" is translated by the *Kana-Kanji* translation method. The number string "2030" expresses "かんせん [kansen]", "きんせん [kinsen]" and so on. The number-*Kanji* dictionary has (2030：観戦), (2030：金銭) and so on. The *Kana-Kanji* dictionary has (かんせん：観戦), (きんせん：金銭) and so on. In the number-*Kanji* translation, "観戦" and "金銭" are able to be applied to "2030". In the *Kana-Kanji* translation, "観戦" is able to be applied to "かんせん". However, "金銭" is not able to be applied. It shows that number strings are more ambiguous than *Kana* strings. Then, the number of key presses decreases using the *Kana-Kanji* translation method for the proofreading in IL-NKT.

4 EVALUATION EXPERIMENT

One system based on IL-NKT has been developed for an experiment. The other system based on LC-KKT uses MS-IME2000* for the translation. LC-KKT is the *Kana-Kanji* translation method using the letter cycling input which is a general method for input of Japanese sentences on mobile phones. We evaluate the number of key presses in these methods.

4.1 EXPERIMENT DATA

The data for the experiment consists of Japanese sentences of the first author's e-mail. The number of inputted characters is 50,000 for the data. There are many kinds of fields and a target field changes frequently in the data.

4.2 EXPERIMENT PROCEDURE

The number strings for the experiment data are translated on IL-NKT and the *Kana* strings are translated on LC-KKT every string. If the translation result has errors, they are proofread by a user. We evaluate the number of key presses to input the correct Japanese sentences on these methods.

4.2.1 KANA-KANJI TRANSLATION METHOD USING LETTER CYCLING INPUT

A user inputs a *Kana* sentence by the letter cycling input method. We express the number of key presses for the input IN_k. The inputted *Kana* sentence is translated into the *Kanji-Kana* mixed sentence. If the *Kana-Kanji* translation result has errors, they are proofread by the user. The proofreading process consists of the judgment of the translation result and the *Kana-Kanji* re-translation. The erroneous characters are chosen by the user and translated again. The user repeats the proofreading process while the translation result has errors. The user inputs a next sentence after the proofreading. We express the number of key presses for the proofreading PR. Then, the number of key presses in LC-KKT is expressed as NP_{LC} and defined as:

$$NP_{LC}(NC) = IN_k(NC) + PR(NC) \qquad (3)$$

*Microsoft Corp. has developed MS-IME2000.

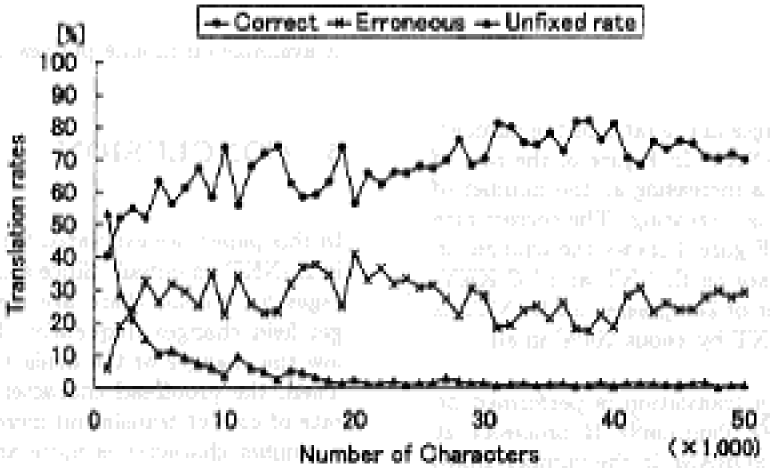

Figure 3: Translation Rates in IL-NKT

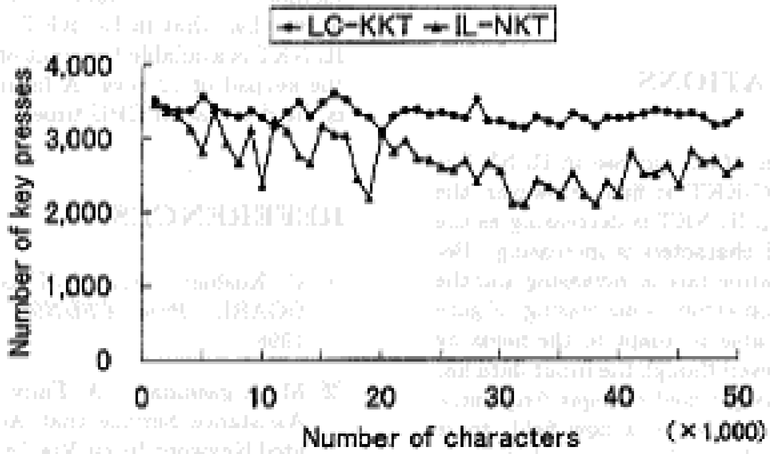

Figure 4: Changes in Number of Key Presses

where NC is the number of inputted characters. For example, the number of key presses is 2,932 for the input of 1,000 *Kana* characters and 420 for the proofreading of the translation result. Then, NP_{LC} is 3,352.

4.2.2 NUMBER-KANJI TRANSLATION METHOD USING INDUCTIVE LEARNING

A user inputs a number string by the degenerated input method. We express the number of key presses for the input IN_n. If the translation result has errors, they are proofread. The erroneous characters are chosen by the user. The chosen characters are translated by LC-KKT. Then, the number of key presses for the proofreading is NP_{LC}. Therefore, the number of key presses in IL-NKT is expressed as NP_{IL} and defined as:

$$NP_{IL}(NC) = IN_n(NC) + NP_{LC}(NC_e) \qquad (4)$$

where NC_e is the number of erroneous characters in the number-*Kanji* translation result. For example, the number of key presses is 1,000 for the input of 1,000 characters and 1,359 for the proofreading of the translation errors whose number is 320. Then, NP_{IL} is 2,359.

4.3 RESULTS

Figure 3 shows the changes in the rates of the number-*Kanji* translation in IL-NKT. In Figure 3, the rate of the correct translation is increasing as the number of the inputted characters is increasing. The correct rate is about 75[%] finally. Figure 4 shows the changes in the number of key presses on IL-NKT and LC-KKT. In Figure 4, the number of key presses on IL-NKT is less than that on LC-KKT by about 20[%] in all.

The number-*Kanji* translation is performed on a computer with Intel® Pentium® II processor at 450MHz and 320MB of SDRAM. In the number-*Kanji* translation from 49,000 to 50,000 characters of the input data, the rate of CPU and RAM used by the system were about 98[%] and 2.8[%], and the time was about 1.5 hours.

4.4 CONSIDERATIONS

In Figure 4, the number of key presses in IL-NKT is not less than that in LC-KKT at first. However, the number of key presses in IL-NKT is decreasing as the number of the inputted characters is increasing. Because the correct translation rate is increasing and the number of the translation errors is decreasing. Figure 3 shows it. IL-NKT is able to adapt to the fields by its own learning ability even though the input data has various fields and the target field changes frequently. When the target field changes to a new field, there are segments unregistered into the segment dictionary in the sentences of the new field. Since the system based on IL-NKT acquires the unregistered segments, the segments are able to be applied in the next translation for the field. It shows that the system based on IL-NKT follows the changes in the fields even though e-mail has many fields. Thus, it is proved that the number of key presses is decreasing by the adaptability of IL-NKT.

The size of the proofreading dictionary influences the correct rate of the *Kana-Kanji* translation usually. When the size is big, the correct rate is high. However, the size of the proofreading dictionary is limited on mobile phones. Then, the correct rate of the *Kana-Kanji* translation on mobile phones is lower than that of MS-IME2000 generally. The correct rate influences $NP_{LC}(NC)$ more than $NP_{LC}(NC_s)$ because NC is more than NC_s in equation (3)(4). Then, IL-NKT is effective all the more when the correct rate of the *Kana-Kanji* translation is low. It shows that IL-NKT is available on mobile phones limited memory.

*Intel® and Pentium® are registered trademarks of Intel Corp.

5 CONCLUSION

In this paper, we evaluated the number of key presses in IL-NKT on e-mail. Since a user inputs various messages for e-mail, there are many fields and the target field changes frequently. IL-NKT is able to follow the changes of the fields by its own adaptability. Then, the proofread characters decrease because the rate of correct translation increases in IL-NKT. Since a number character is more ambiguous than a *Kana* character, the number of the segment candidates on the number-*Kanji* translation is more than that on the *Kana-Kanji* translation. The proofreading on IL-NKT is performed using the *Kana-Kanji* translation method. Then, the number of key presses in IL-NKT is less than that in LC-KKT. Thus, it is proved that IL-NKT is available for input of sentences for e-mail on the keypad of 12 keys. A future problem for IL-NKT is the decrease of CPU time.

REFERENCES

[1] C. Kushler, AAC USING A REDUCED KEYBOARD, *Proc. CSUN98*, Los Angeles, March 1998.

[2] M. Higashida, A Fully Automated Directory Assistance Service that Accommodates Degenerated Keyword Input Via Telephone, *Proc. PTC'97*, Honolulu, January 1997, 167–174.

[3] K. Araki, Y. Momouchi and K. Tochinai, Evaluation for adaptability of Kana-Kanji translation of non-segmented Japanese Kana sentences using inductive learning, *Conference Working Papers of PACLING-II*, Brisbane, April 1995, 1–7.

[4] M. Matsuhara, K. Araki, Y. Momouchi and K. Tochinai, Evaluation of Number-Kanji Translation Method of Non-Segmented Japanese Sentences Using Inductive Learning with Degenerated Input, *Proc. AI'99*, Sydney, December 1999, 474–475.

[5] M. Nagao and S. Mori, A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese, *Proc. COLING94*, Kyoto, August 1994, 611–615.