MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

An Efficient OCR Error Correction Method for Japanese Text Recognition

Toru Hisamitsu*, Katsumi Marukawa, Yoshihiro Shima**, Hiromichi Fujisawa** and Yoshihiko Nitta***

* Advanced Research Laboratory, Hitachi, Ltd., Hatoyama, Saitama 350-03, JAPAN
** Central Research Laboratory, Hitachi, Ltd., Kokubunji, Tokyo 185, JAPAN
*E-mail: {hisamitu, nitta}@harl.hitachi.co.jp
**E-mail: {marukawa, shima, fujisawa}@crl.hitachi.co.jp

Abstract

OCR error correction using Japanese morphological analysis contains two time-consuming procedures: extraction of candidate words from combinations of candidate characters, and finding the most plausible word sequence in combinations of the candidate words. In this paper an optimal word extraction technique, and the use of lexical entries that are tailored for Japanese verb inflection, are investigated and developed. Compared to a standard method, the new method requires 84% less computation, and captures 2.6% more candidate words. The new design of lexical entries reduces the chart parsing computation by 20%. The error correction rate of the system is 86.9%, which is 19.6% higher than that of the standard one.

1 Introduction

In ordinary Japanese written sentences, words are not separated by spaces. Thus error correction algorithms for Japanese text recognition using Japanese morphological analysis is forced to use two time-consuming but indispensable procedures: (1) Searching for lexical entries in combinations of characters arranged in a "candidate character lattice" (a sequence of lists of candidate characters output by a character recognition module) and extracting candidate words from a dictionary at each position in the output character string, (2) Finding the most plausible word sequence from the "candidate words lattice" (the set of words extracted by procedure (1)).

Those procedures take most of the error correction time. Procedure (1) consumes more than half of the processing time [1]. Thus, a more efficient word extraction method would be helpful. At the same time, word extraction must be sufficiently precise so that as many words as possible are found. This is due to missing words resulting in linguistic post processing failure, which consequently results in a lower error correction rate.

This paper investigates an optimal word extraction technique and the use of lexical entries specially tailored for Japanese verb inflection. Compared to the standard method $\mathbf{Ext^1}(c_1)$ (mentioned in section **2**), this method requires significantly less computation and attains a higher error correction rate even if the recognition rate of the first candidate character is quite low. The comparative results of experiments will also be shown.

2 Existing Word Extraction Methods

Conventional extraction methods for words longer than two characters are classified into two groups: Candidate Characters Driven methods (**CCDM**s) and Dictionary Driven methods (**DDM**s).

CCDMs generate lexical entries of the dictionary by generating combinations of characters in a candidate character lattice. **CCDM**s have several deficiencies: 1) they can only extract words whose characters completely appear in the candidate character lattice; 2) latent long words in the candidate character lattice may result in seriously low computational efficiency, even if the dictionary has a carefully designed TRIE structure; 3) low recognition rate directly results in low *word capturing rate* and low error correction rate.

DDMs were developed to resolve the deficiencies of **CCDM**s. **DDM**s use a dictionary in which each word can be accessed by one of the characters it contains as a key. In other words, for given natural numbers $\{i_1, i_2, \ldots, i_n\}$, if the i_k-th character in a word is contained in the candidate character lattice, the word is extracted from the dictionary. The words extracted in this way are referred to as *pre-candidate words*. Next **DDM**s compare the words with the candidate character lattice. The comparing procedure is referred to as a *matching procedure*. In this paper, $\mathbf{Ext}(c_{i_1} \vee \cdots \vee c_{i_n})$ denotes the aforementioned word extraction method, and $\mathbf{AVM}\{\mathbf{E}(c_{i_1} \vee \cdots \vee c_{i_n})\}$ denotes the average number of the matching procedures of $\mathbf{E}(c_{i_1} \vee \cdots \vee c_{i_n})$ at each position. If one of the pre-candidate words satisfies certain conditions, then the word is registered in the *candidate word lattice*.

Unlike **CCDM**s, **DDM**s can recover characters which do not appear in the candidate character lattice. For this reason, **DDM**s have been widely used in OCR error correction research [1][2].

However, conventional **DDM**s still have several problems. In terms of efficiency, even the $\mathbf{AVM}\{\mathbf{Ext}(c_1)\}$ amounts to a large number. The use of $\mathbf{Ext}(c_1)$ is therefore usually restricted to only extracting words whose first characters match the top candidate character at a position, and comparing those words with the candidate character lattice. We refer to this restricted $\mathbf{Ext}(c_1)$ as $\mathbf{Ext^1}(c_1)$, which has been used as a standard **DDM**. However, $\mathbf{AVM}\{\mathbf{Ext^1}(c_1)\}$ is reported to amount to about 50. Moreover, this restriction results in a low *word capturing rate* (we refer to *word capturing rate* as the ratio between the number of correct candidate words in *candidate word lattice* and the actual number of words). As a consequence, a lower error correction rate occurs.

$\mathbf{Ext^1}(c_1 \vee c_2)$, the restricted version of $\mathbf{Ext}(c_1 \vee c_2)$, only uses the top candidate characters. The

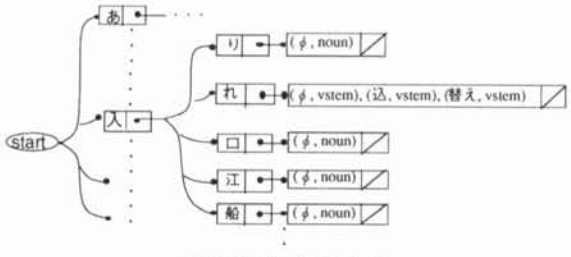
AVM{Ext^1($c_1 \vee c_2$)} value is expected to be about two times that of **AVM{Ext^1(c_1)}**. In addition, a 'weak' condition, in which a word is registered in the candidate word lattice if all characters except one character of the word appear in the candidate character lattice, might pass a large quantity of incorrect words. It may result in replacing correct characters with incorrect ones. Hence, it is difficult to improve error correction efficiency while preserving error correction rate when one uses conventional **DDM**s.

3 Proposed Word Extraction Method

3.1 The Method

In this section a simple method to extract candidate words is introduced. The method uses a dictionary in which each word can be accessed by using its first L characters as a key. This method can be classified as a type of **DDM**. This method equals **Ext**(c_1) if $L = 1$, and equals **CCDM** for a sufficiently large L. In this paper this method is referred to as **Ext**($c_1 \wedge \cdots \wedge c_L$). The dictionary for **Ext**($c_1 \wedge c_2$) is illustrated in **Figure 1**.

This section compares three methods for word extraction (**Ext^1**(c_1), **Ext^1**($c_1 \vee c_2$), and **Ext**($c_1 \wedge c_2$)) and discusses an optimal L for Japanese text recognition.



Dictionary for Ext(c1∧c2)

Figure 1
Illustration of Dictionary Data Structure

3.2 Estimating *Word Capturing Rate*

First consider the probability that a word $w = c_1 c_2 \ldots c_m$ (m ≥ 2) in an input sentence can be captured in the candidate words. The following notations are necessary:

$p_M(c)$: the probability that character c is recognized by the M-th candidate,

$P_{w;\,M}$: the probability that every c_i is recognized by the M-th candidate,

$P_{w;\,1,\,1}$: the probability that c_1 is recognized as the top candidate,

$P_{w;\,1\wedge 2,\,M}$: the probability that both c_1 and c_2 are recognized by the M-th candidate,

$P_{w;\,1\vee 2,\,M}$: the probability that c_1 or c_2 is recognized by the M-th candidate.

Using these notations, it can be shown that the following equations hold:

$$P_{w;\,M} = p_M(c_1) \cdot p_M(c_2) \cdot \cdots \cdot p_M(c_m),$$

$$P_{w;\,1,\,1} = p_1(c_1),$$

$$P_{w;\,1\wedge 2,\,M} = p_M(c_1) \cdot p_M(c_2),$$

$$P_{w;\,1\vee 2,\,M} = 1 - (1 - p_M(c_1)) \cdot (1 - p_M(c_2)).$$

For simplicity, assume that characters recognized by the M-th candidate are all used to extract candidate words (see NOTES). The condition which selects words to be registered is defined as:

> A word is not registered in the candidate word lattice if equal or more than two characters are missed in the character candidate lattice.

Under these conditions the following equivalence relations hold:

$P_{w;\,M} \Leftrightarrow$ the probability that w is captured in the candidate word lattice by using a **CCDM**,

$P_{w;\,1,\,1} \Leftrightarrow$ the probability that w is captured in the candidate word lattice by using **Ext^1**(c_1),

$P_{w;\,1\wedge 2,\,M} \Leftrightarrow$ the probability that w is captured in the candidate word lattice by using **Ext**($c_1 \wedge c_2$).

$P_{w;\,1\vee 2,\,1} \Leftrightarrow$ the probability that w is captured in the candidate word lattice by using **Ext^1**($c_1 \vee c_2$).

To compare these probabilities, it is assumed that $p_M(c) = p_M$ (i.e., $p_M(c)$ is independent of each character c). Since we are interested in a situation where the recognition rate is low, the probabilities when M=5, $p_1 = 0.90$ and $p_5 = 0.98$ are calculated. Since $P_{w;\,M} = p_1^{|w|}$, $P_{w;\,M}$ = 0.960, 0.941, and 0.922 for $|w|$ is 2, 3, and 4, respectively, where $|w|$ denotes the length of w. On the other hand, $P_{w;\,1,\,1}$(= 0.90), $P_{w;\,1\wedge 2,\,M}$ (=0.960) and $P_{w;\,1\vee 2,\,M}$ = 0.99 are independent of the length of w.

From the above calculation, it can be expected that word extraction using **Ext^1**($c_1 \vee c_2$) attains the highest word capturing rate and **Ext**($c_1 \wedge c_2$) the second highest. The results also show that both **CCDM** and **Ext^1**(c_1) are problematic for use in precise word extraction.

Note that the high word capturing rate of **Ext^1**($c_1 \vee c_2$) does not necessarily result in precise error correction. This is because, as mentioned in section **2**, **Ext^1**($c_1 \vee c_2$), with the aforementioned weak condition, may register a large number of candidate words. It may also cause error replacement of correct characters. This is shown experimentally in section **5**.

3.3 Estimating the Number of Matching Procedures

The number of matching procedures at each position is another important measure in comparing word extraction methods, as word matching procedures occupy half of the post processing time.

An experiment showed that **AVM(Ext^1**(c_1)) = 47.1, **AVM(Ext^1**($c_1 \vee c_2$)) = 121.9 and **AVM(Ext**($c_1 \wedge c_2$)) = 7.54 (see **Fig.1** and **NOTES**). According to this, **Ext**($c_1 \wedge c_2$) is the best of the three as it reduces the matching procedures of the standard method **Ext^1**(c_1) by 84%. **Ext^1**($c_1 \vee c_2$) is considerably worse even than the standard method **Ext^1**(c_1).

When $L \geq 3$, we cannot expect to reduce the number of matching procedures, since the number of combinations

of candidate characters increases exponentially as L grows. The word capturing rate also decreases exponentially. For example, $P_{w,\ 1\wedge 2\wedge 3,\ M}$ equals 0.941 under the same conditions mentioned in **3.1**. In addition, if $L = 2$, it is large enough to directly cover the largest part of Japanese words, which consists of two letter words.

In light of the above, it was found that $\mathbf{Ext}(c_1 \wedge c_2)$ is the optimal dictionary-driven word extraction technique.

4 Newly Devised Lexical Entries for Japanese Verb Inflection

Another important, but time-consuming part of OCR error correction is to find a feasible word sequence from the candidate words. The process is the same as the extraction of feasible word sequences in Japanese morphological analysis. Various techniques in Japanese morphological analysis can thus be applied to the process.

A rule-based heuristic for selecting a feasible word sequence was employed, which had already been devised and introduced[3]. It is simple, portable, and accurate.

In the system, a new lexical treatment of Japanese verb inflection is applied. This treatment is developed to improve the efficiency of ordinary morphological analysis [4] and also to be effective for OCR post-processing.

This treatment can be described using a concrete example. The character string "消さなかった(*kesanakatta*: did not extinguish)" is phonologically analyzed as "kes (to extinguish) + ana (Negative) + katta (Past)". However, as phonological morphemes such as '*kes*' and '*ana*' are not observable in Japanese character strings, the head vowel of an inflectional affix has been attached to the tail of a consonant verb stem in the standard verb inflection description. This generates an observable 'inflected form' (*Katsuyoukei*). Thus,

'*kes*' + '*ana*' ---> '*kesa*' + '*na*' ='けさ'+ 'な'.

As a result, a consonant stem verb is considered to have several inflected forms, such as {けさ(*kesa*), けし(*kesi*), けす(*kesu*), けせ(*kese1*), けせ(*kese2*), けそ(*keso*)}.

The most common treatment of such inflected forms involves separating 'inflectional endings' (the last *hiragana* of each inflected form) from the inflected forms and registering them as lexical entries (see **Table 1**).

The same example can now be analyzed as follows:

消さなかった ----> 消 / さ / な / かっ /た

kesanakatta *ke*[*s*]: to extinguish / *sa*: ϕ / [*a*]*na*: Neg./ kat: ϕ /*ta*: Past

However, analysis by this method requires more segmentation than phonological analysis, and results in lower efficiency. To resolve this problem, the consonant '*s*' is attached to the head of '*ana*', and an entry, 'さな (*sana* =*s*+*ana*)', is generated as a kind of an allomorph of '*ana*'. At the same time, the stem '消' is marked as a morpheme which can only be followed by "*s*-attached inflectional affixes," that is, {させ (*s*+*ase*: Causative), され (*s*+*are*: Passive), さな (*s*+*ana*: Negative,....}. Other lexical entries are generated in the same manner (see **Table 2**). The previous example can be analyzed as follows:

消さなかった (*kesanakatta*) ---> 消 / さな / かった

It has proved that this method minimizes the number of lexical entries and reduces chart parsing computation. Since error correction of Japanese text recognition strongly depends on Japanese morphological analysis, it can be expected that the new design improves the efficiency of error correction of Japanese text recognition. To estimate the efficiency of the chart parsing, the average number of tests per position is introduced, which tests check for the connectability between 'partial-solution fragments', and candidate words in the candidate word lattice (for 'partial-solution fragments' and the details on the chart parser, see [4]). This measure reflects the chart parsing time and is independent of implementation variations. The comparison is shown in the next section.

entry	comments
消	stem
：	
さ	*Mizenkei* inflectional ending1
し	*Renyookei* inflectional ending
す	*Rentaikei* inflectional ending
す	*Shuushikei* inflectional ending
せ	*Kateikei* inflectional ending
せ	*Meireikei* inflectional ending
そ	*Mizenkei* inflectional ending2
：	
な	Negative
せ	Causative
れ	Passive
た	Past
：	

Table 1
Examples of Lexical Entries (**ST**)

entry	comments
消	stem
：	
さな	Negative (s + *ana*)
させ	Causative (s + *ase*)
され	Passive (s + *are*)
した	Past (s + *ita*)
：	

Table 2
Examples of Lexical Entries (Proposed Method)

5 Experimental Results

In the experiments, sample sentences containing about 25,000 characters taken from the *Nikkei Shinbun* (a Japanese newspaper) were used. The recognition rate was relatively low ($\mathbf{ARR(1)} = 0.913$ and $\mathbf{ARR(5)} = 0.9821$, where $\mathbf{ARR}(k)$ denotes the accumulated recognition rate until the k-th candidate character).

Several methods were compared using four measures:

(1) The average number of the word matching procedures (mentioned in section **2**),

(2) The word capturing rate (mentioned in section **2**),

(3) The average number of tests which check for the connectability between partial-solution fragments and candidate words in the candidate word lattice (mentioned in the previous section), and

(4) The error correction rate, which is defined as the value (C-B) / (C+D+E), where B denotes the number of correct characters that were replaced with erroneous characters, C denotes the number of erroneous characters that were successfully replaced with correct characters, D denotes the number of erroneous characters that were replaced with other erroneous characters, and finally, E denotes the

number of erroneous characters that were not changed.

The experimental data can be summarized with four points:

1) The average number of word matching of the method was 7.5. This is 84% less than that obtained by the standard method $\mathbf{Ext^1}(c_1)$ (see **Fig.2**),

2) The *word capturing rate* of the method was 98.8%. This is 2.6% higher than the rate obtained by $\mathbf{Ext^1}(c_1)$ (see **Table.3**),

3) The new lexical entry design, using $\mathbf{Ext}(c_1 \wedge c_2)$, reduced chart parsing computation by 20% compared to the standard lexical entry design using $\mathbf{Ext}(c_1 \wedge c_2)$, and

4) The error correction rate of the proposed method was 86.9%. This is 19.6% higher than the rate of $\mathbf{Ext^1}(c_1)$ using standard lexical entries, and also 3.5% higher than the rate of $\mathbf{Ext}(c_1 \wedge c_2)$ using standard lexical entries (see **Fig.4**).

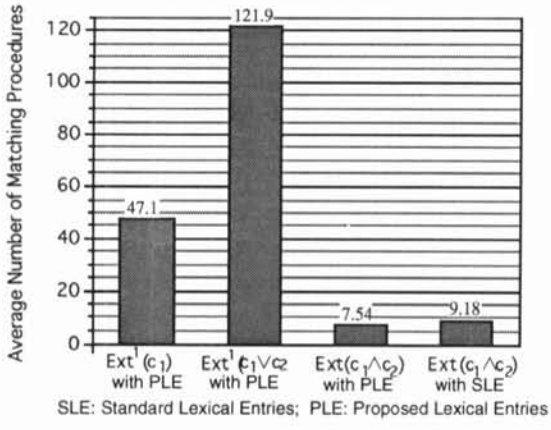


Figure 2
Comparison of the Average Number of Matching Procedures

Method	$\mathbf{Ext^1}(c1 \vee c2)$		$\mathbf{Ext^1}(c1)$		$\mathbf{Ext}(c1 \wedge c2)$	
	SLE	PLE	SLE	PLE	SLE	PLE
Word Capturing Rate	98.33%	98.89%	96.11%	95.45%	98.68%	99.07%

Table 3
Comparison of Word Capturing Rate

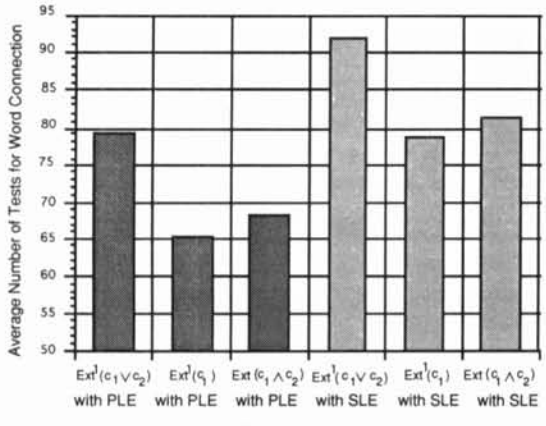


Figure 3
Comparison of Parsing Efficiency

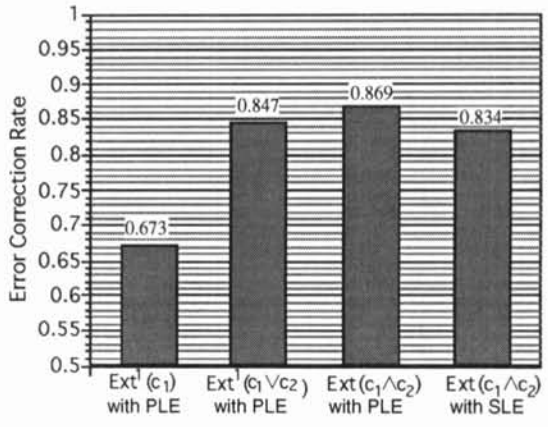


Figure 4
Comparison of Error Correction Rate

6 Conclusion

In this paper a newly developed optimal word extraction technique, and the use of lexical entries specially tailored for Japanese verb inflection, were introduced. Compared to the standard methods, the new approach requires significantly less computation and attains higher error correction rates. These rates are obtained even if the recognition rate of the first candidate character is relatively low.

NOTES

1) In the experiments reported in **section 5**, M was not fixed, but a type of neighborhood of top candidates was employed. An element c_i of the neighborhood satisfies two conditions:

(1) $CR(c_1) - CR(c_i) \leqq \alpha_1$,

(2) $CR(c_{i-1}) - CR(c_i) \leqq \alpha_2$,

where α_1 and α_2 are fixed positive numbers, and $CR(c_i)$ denotes the confidence rate of the i-th candidate character. In the experiments, the number of characters in the neighborhood was 3.9 per position, on average.

References

[1] T. Takao et al., "Implementation and Evaluation of Post Processing for Japanese Document Readers". *Trans. of IPSJ*, Vol.30, No. 11, pp.1394-1401 (1989).

[2] T. Sugimura, "Error Correction Method for Character Recognition Based on Confusion Matrix and Morphological Analysis", *Trans. of IEICE*, Vol.J72-D-II, No.7, pp.993-1000 (1989).

[3] T. Hisamitsu et al., "A Generalized Algorithm for Japanese Morphological Analysis and Comparative Estimation of Some Heuristics", *Trans. of IEICE*, Vol.J77-D-II, No.5, pp.959-969 (1994).

[4] T. Hisamitsu et al., "An Efficient Treatment of Japanese Verb Inflection for Morphological Analysis", *Proc. of COLING'94 Kyoto*, pp.194-200 (1994)