Kana-Kanji Conversion System with Input Support Based on Prediction

Yumi ICHIMURA, Yoshimi SAITO, Kazuhiro KIMURA and **Hideki HIRAKAWA**

Human Interface Laboratory
Corporate Research & Development Center, TOSHIBA Corp.
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki-shi,
Kanagawa, 212-8582 Japan
yumi.ichimura@toshiba.co.jp

Abstract

We propose a *kana-kanji* conversion system with input support based on prediction. This system is composed of two parts: prediction of succeeding *kanji* character strings from typed *kana* ones, and ordinary *kana-kanji* conversion. It automatically shows candidates of *kanji* character strings which the user intends to input. Our prediction method features: (i)Arbitrary positions of typed *kana* character strings are regarded as the top of words. (ii)A system dictionary and a user dictionary are used, and each entry in the system dictionary has *certainty factor* calculated from the frequency of words in corpora. (iii)Candidates are estimated by *certainty factor* and *usefulness factor*, and likely ones with greater factors than thresholds are shown. The proposed system could reduce the user's key input operations to 78% from the original ones in our experiments.

1 Introduction

TOSHIBA developed the world's first Japanese word processor in 1978. Unlike languages based on an alphabet, Japanese uses thousands of *kanji* characters of varying complexity. Hence, to arrange all of *kanji* characters on keyboard is difficult. On the other hand, *kana* characters which are phonetic scripts of Japanese have 83 variations; these can be arranged on keyboard. As a result, conversion from *kana* notations to *kanji* ones, what is called *kana-kanji* conversion, has been used. Since Japanese is not written separately by words, segmentation of typed *kana* character strings has ambiguity. And an ambiguity in conversion exists, too; a *kana* notation may correspond to some different *kanji* notations. These make *kana-kanji* conversion challenging.

We have made efforts to raise a precision of *kana-kanji* conversion, thinking that high precision can provide a better input environment for the user. A precision of our *kana-kanji* conversion system reaches 95-98% for several kinds of texts in our previous experiments. Nevertheless, this approach is not enough in the situations where fast typing is hard, e.g., for beginners who are not familiar with keyboard or for palm-size computers. Thus, new method to reduce key input operations is needed.

We propose a *kana-kanji* conversion system with input support based on prediction. This system is composed of two parts: prediction of succeeding *kanji* character strings from typed *kana* ones, and ordinary *kana-kanji* conversion. It automatically shows candidates of *kanji* character strings which the user intends to input. The candidates change as the user inputs *kana* characters. If no appropriate choice is presented, the candidates automatically disappear when the next *kana* character is entered. Our system, therefore, can be used in the same manner as an ordinary *kana-kanji* conversion system, and allows the user to save time and efforts for key input without learning new key operations.

We have been considered two issues to generate accurate candidates:

(i) How to determine where typed *kana* character strings are segmented; since Japanese is not written separately by words, determination of positions where words start is needed to retrieve dictionaries.

(ii) How to determine when prediction candidates are presented; if all of retrieval results are always shown, a system cannot be convenient.

Surveying previous works from the view on above issues, we found that the Reactive Keyboard has been proposed (Darragh et al., 1980). It accelerates typewritten communication with

a computer system by predicting what the user is going to type next. In this system, the top of typed character strings is regarded as the top of words, because English is written separately by words; the issue of segmentation of character strings does not occur.

On the other hand, in previous works for Japanese, a predictive pen-based text input method has been proposed (Fukushima and Yamada, 1996). In this system, character strings are input by hand-writing on LCD panel. Since the user usually inputs not only by *kana* but also by *kanji* and an alphabet, entered character strings are segmented with the help of the variety of characters. Thus, the issue of segmentation is not considered.

The POBox (Pen-Operations Based On eXample) which is a text input method for pen-based computers has also been proposed (Masui, 1998). It shows succeeding candidates from character strings input by software keyboard. Arbitrary positions of input character strings can be regarded as the top of words, and retrieval results are always shown as candidates; the prediction ordering is based on the user's previous choice. Since input speed by pen is not faster than that by keyboard, time to choose candidates is shorter than that to input characters. Hence, even if many candidates are shown, this method is effective for pen-based computers. It is, however, inefficient for ordinary keyboard.

We propose a system with following features:

(i) Arbitrary positions of typed *kana* character strings are regarded as the top of words.

(ii) A system dictionary and a user dictionary are used, and each entry in the system dictionary has *certainty factor* calculated from the frequency of words in corpora.

(iii) Candidates are estimated by *certainty factor* and *usefulness factor*, and likely ones with greater factors than thresholds are shown.

These features provide an efficient Japanese text input environment for ordinary keyboard without learning new key operations.

Section 2 shows an example of text input using the proposed system. Section 3 explains an input support method based on prediction. Section 4 shows efficiency of our system by means of experiments. Section 5 describes conclusions.

2 Example of Text Input

Figure 1 shows an example of text input using the proposed system. Suppose that the user intends to input a sentence "下記の会議を開催しますのでご参集願います (we request your attendance at the following meeting)", typing *kana* characters "かきのかいぎをかいさいしますのでごさんしゅうねがいます (*kakino kaigiwo kaisai shimasunode gosanshuu negaimasu*)". When the user types "か (*ka*)", "き (*ki*)" , and "の (*no*)" keys, the system automatically opens a prediction menu window just below the typed characters, and shows two candidates in the menu window (Fig.1(a)):

下記の住所にささやかながら
(at the following address, modest ...)
下記の住所にささやかな
(at the following address, modest ...)

The first candidate is highlighted. If the menu window contains an appropriate candidate, the user can choose it by cursor; otherwise the user can continue entering the next characters. Subsequently, when "か (*ka*)" key is typed, the prediction menu window disappears (Fig.1(b)). When "い (*i*)" and "ぎ (*gi*)" keys are typed, the system automatically opens a prediction menu window again, and shows four new candidates (Fig.1(c)):

会議を開催しますのでご参集願います
(we request your attendance at the meeting)
会議を開催します
(we hold the meeting)
会議を行います
(we hold the meeting)
会議を執り行います
(we hold the meeting)

Here the first one is what the user just wants; the user enters select key, then the prediction menu window disappears, and chosen candidate is inserted in the edit area. If remaining *kana* character string which was not included in the chosen candidate exists, *kana-kanji* conversion starts automatically; the first three *kana* characters of this sentence "かきの (*kakino*)" is converted to *kanji* notation "下記の (the following)" (Fig.1(d)). This is the first result of *kana-kanji* conversion, so that the user can change it to others. An overline of the conversion result in Fig.1(d) shows that this result is not fixed yet.

In above example, while 27 *kana* characters are needed to input in ordinary *kana-kanji* con-

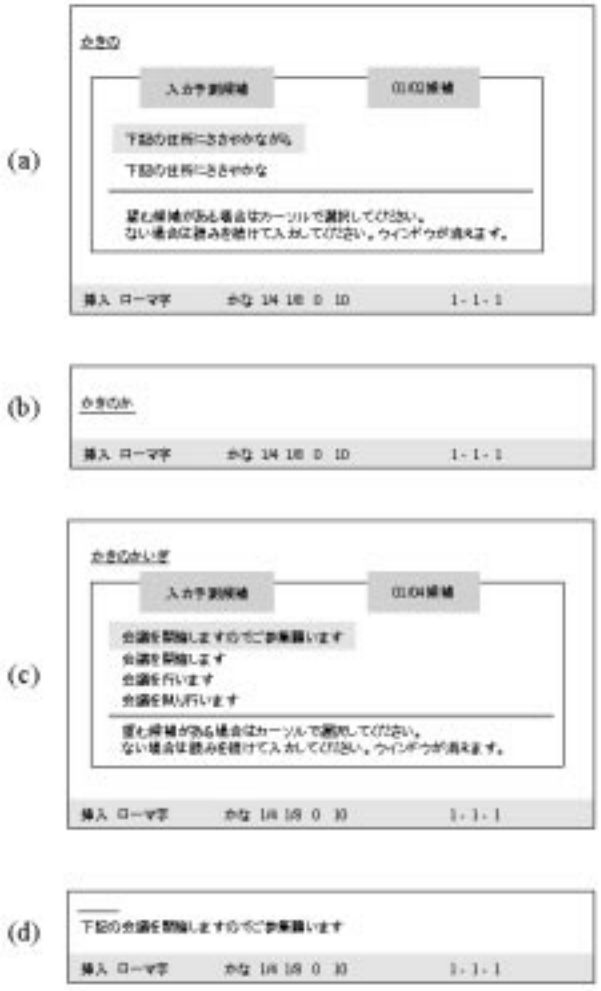

Figure 1: Example of Japanese text input using word processor with input support. (a)“か”, “き”, and “の” keys are typed. (b)“か” key is typed, subsequently. (c)“い” and “ぎ” keys are typed, subsequently. (d)The first candidate in (c) is chosen.

version, our system can reduce the input of 21 *kana* characters, “をかいさいしますのでごさんしゅうねがいます (*wo kaisai shimasunode gosanshuu negaimasu*)”; only 6 *kana* characters are needed to input.

3 Input Support Method Based on Prediction

In this section, an overview of the system is shown. Then dictionaries used in the system, factors for estimation of candidates, and user learning are described.

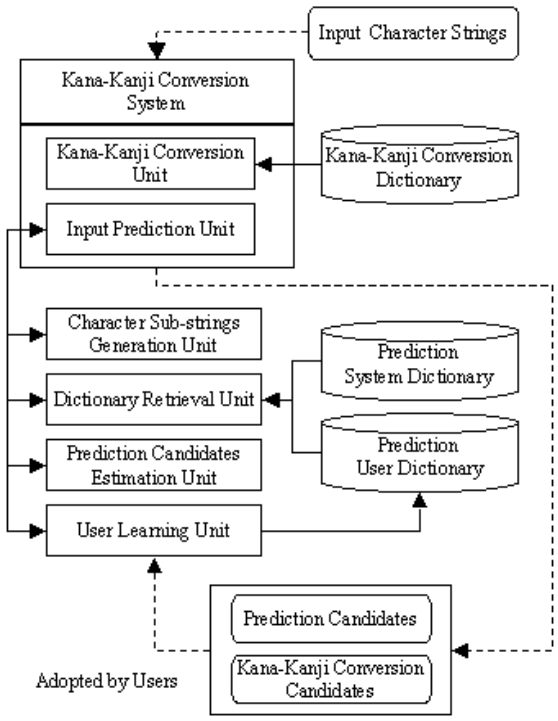

Figure 2: Diagram of the proposed system.

3.1 Overview of the system

Figure 2 shows a diagram of the proposed system. It is composed of a *kana-kanji* conversion unit and an input prediction unit, and the latter has following four sub-units:

Character Sub-strings Generation Unit(a) generates character sub-strings obtained from segmentation of typed *kana* character strings.

Dictionary Retrieval Unit(b) retrieves prediction dictionaries using character sub-strings generated by Unit(a).

Prediction Candidates Estimation Unit(c) calculates *certainty factor* and *usefulness factor* for all of retrieved results by Unit(b) to estimate candidates.

User Learning Unit(d) extracts phrases adopted by the user, and automatically registers them into the user dictionary.

3.2 Prediction Dictionary

Two kinds of dictionaries are used as a prediction source:

(i) System Dictionary consists of high frequent phrases.

(ii) User Dictionary consists of phrases learned from texts which the user typed before.

Each dictionary includes words and phrases without distinction. This is because Japanese is not written separately by words, and high frequent phrases consist of various grammatical forms, such as single word or two words. And each entry has *kana* notation (phonetic script) and *kanji* one.

3.3 Estimation of Prediction Candidates

Two kinds of factors are used to estimate candidates:

(i) Certainty Factor indicates how certain a candidate is.

(ii) Usefulness Factor indicates how useful a candidate is.

These two factors vary as the user inputs a character. Retrieval results are sorted in order of these factors, and only ones with greater factors than thresholds are shown as candidates.

3.4 Calculation of Certainty Factor

Certainty factors for entries in the system dictionary and the user dictionary are calculated in different manner.

First we make some notational conventions. A typed *kana* character string is denoted by S, which has right sub-strings S_i $(1 \leq i \leq L(S))$. $L(x)$ is the length of a string x. An entry in the dictionary is denoted by W, which has *kanji* notation W_H and *kana* notation W_Y.

3.4.1 Entry of System Dictionary

When S is typed, *certainty factor* for W in the system dictionary is calculated as follows:

$$Certainty\ factor(W|S) =$$

$$\begin{cases} \frac{F_H(W_H)}{F_K(S_i)}, & \text{when } S \text{ has a right sub-string } S_i \\ & \text{which partially matches with the} \\ & \text{head of } W_Y \\ 0, & \text{otherwise} \end{cases}$$

where $F_H(W_H)$ is the frequency of W_H in *kanji* notation corpus, and $F_K(S_i)$ is the frequency of S_i in *kana* notation corpus corresponding to *kanji* one.

For example, *certainty factor* for "かな漢字変換 (*kana-kanji* conversion)" is calculated using the frequency in Table 1:

Table 1: Frequency for "かな漢字変換" in two corresponding corpora: *kanji* notation corpus with 155,000 characters, and *kana* one with 227,000 characters.

Character strings	Frequency in *kana* notation corpus
か	6,720
かな	191
かなか	114
かなかん	94
かなかんじ	87
かなかんじへ	78
かなかんじへん	77
かなかんじへんか	76
かなかんじへんかん	76
Character strings	**Frequency in *kanji* notation corpus**
かな漢字変換	70

$$Certainty\ factor(\text{かな漢字変換} \mid \text{かな}) = 70/191 = 0.366$$

$$Certainty\ factor(\text{かな漢字変換} \mid \text{かなか}) = 70/114 = 0.614$$

The values of *certainty factor* corresponding to every character sub-strings are described in the system dictionary, and are read out at retrieval.

3.4.2 Entry of User Dictionary

Since the system cannot infer which phrases would be registered into the user dictionary, calculation of *certainty factor* for an entry in the user dictionary from corpora may be impossible. Hence, when S is typed, *certainty factor* for W in the user dictionary is calculated as follows:

$$Certainty\ factor(W|S) =$$

$$\begin{cases} \frac{\alpha}{N(S_i)}, & \text{when } S \text{ has a right sub-string } S_i \\ & \text{which partially matches with the} \\ & \text{head of } W_Y \\ 0, & \text{otherwise} \end{cases}$$

where $N(S_i)$ is the number of entries whose *kana* notations start from S_i in the user dictionary, and α is a constant to give greater factor for entries in the user dictionary than that in the system dictionary; i.e., the user dictionary has priority.

3.5 Calculation of Usefulness Factor

An increase in the length of typed *kana* character strings raises the certainty on prediction, but lessens the usefulness. Hence, *usefulness factor* is introduced in addition to *certainty factor*. When S is typed, *usefulness factor* for W is calculated as follows:

$$Usefulness\ factor(W|S) =$$

$$\begin{cases} L(W_Y) - L(S_i), & \text{when } S \text{ has a right substring } S_i \text{ which partially matches with the head of } W_Y \\ 0, & \text{otherwise} \end{cases}$$

3.6 User Learning

After the user adopts prediction or *kana-kanji* conversion candidates, words with longer length than threshold and phrases which satisfy given grammatical conditions are extracted; these are automatically registered into the user dictionary.

For example, suppose that the user intends to input a phrase "会議に出席する (attend at the meeting)", typing *kana* characters "かいぎにしゅっせきする (*kaigini shusseki suru*)". When "か (*ka*)", "い (*i*)", and "ぎ (*gi*)" keys are typed, four candidates are shown in the prediction menu window (Fig.1(c)). Here the prediction menu window dose not contain a candidate which the user wants, then the user continues entering the next *kana* characters "にしゅっせきする (*ni shusseki suru*)" and *kana-kanji* conversion key. As a result, "かいぎにしゅっせきする (*kaigini shusseki suru*)" is converted to "会議に出席する (attend at the meeting)".

When this conversion candidate is adopted, two words and a phrase are registered into the user dictionary: "会議 (meeting)", "出席する (attend)", and "会議に出席する (attend at the meeting)". If "か (*ka*)", "い (*i*)", and "ぎ (*gi*)" keys are typed after this learning, "会議に出席する" is contained in the prediction menu window.

4 Experiments

Efficiency of the proposed system is shown by means of experiments.

4.1 Evaluation Measures

Neither start key for prediction nor cancel key for prediction candidates are needed. And select key to adopt candidates is needed in both of prediction and ordinary *kana-kanji* conversion; we need not take into account of the input of select key. Hence, the length of complemented *kana* characters is just a decrease in key input operations. Two evaluation measures, an operation ratio and a precision, are defined as follows:

$$Operation\ ratio = \frac{P-Q}{P} \times 100\ (\%)$$

$$Precision = \frac{R}{S} \times 100\ (\%)$$

where P is the length of the original *kana* text, Q is the length of *kana* characters complemented by prediction, R is the number of shown prediction menu windows containing appropriate choices, and S is the number of all of shown prediction menu windows.

4.2 Data and Conditions

Two kind of texts, a paper on natural language processing and a letter, were used in our experiments; these texts were not included in the corpora used to calculate *certainty factor*. A system dictionary with 37,926 entries was used. Thresholds of *certainty factor* and *usefulness factor* were 0.1 and 2. The number of candidates presented in a prediction menu window was five or less. If a prediction menu window contained an appropriate choice, it was always adopted. With a view to examining each contribution of the system dictionary and user learning, experiments were carried out in three cases:

(i) Only the system dictionary was used.

(ii) Only user learning was used.

(iii) Both the system dictionary and user learning were used.

We calculated the length of complemented *kana* characters automatically. An operation ratio and a precision were recorded at every input of 4,500 *kana* characters.

4.3 Results

Figure 3 shows experimental results.

Decrease in key input operations: Using both the system dictionary and user learning, for the paper, an operation ratio was 97.3–78.6% (line (r3) in Fig.3(a)) and a precision was 20.0–26.7% (line (p3) in Fig.3(c)); for the letter, an operation ratio was 80.7–78.1% (line (r3) in Fig.3(b)) and a precision was 26.1–29.6% (line (p3) in

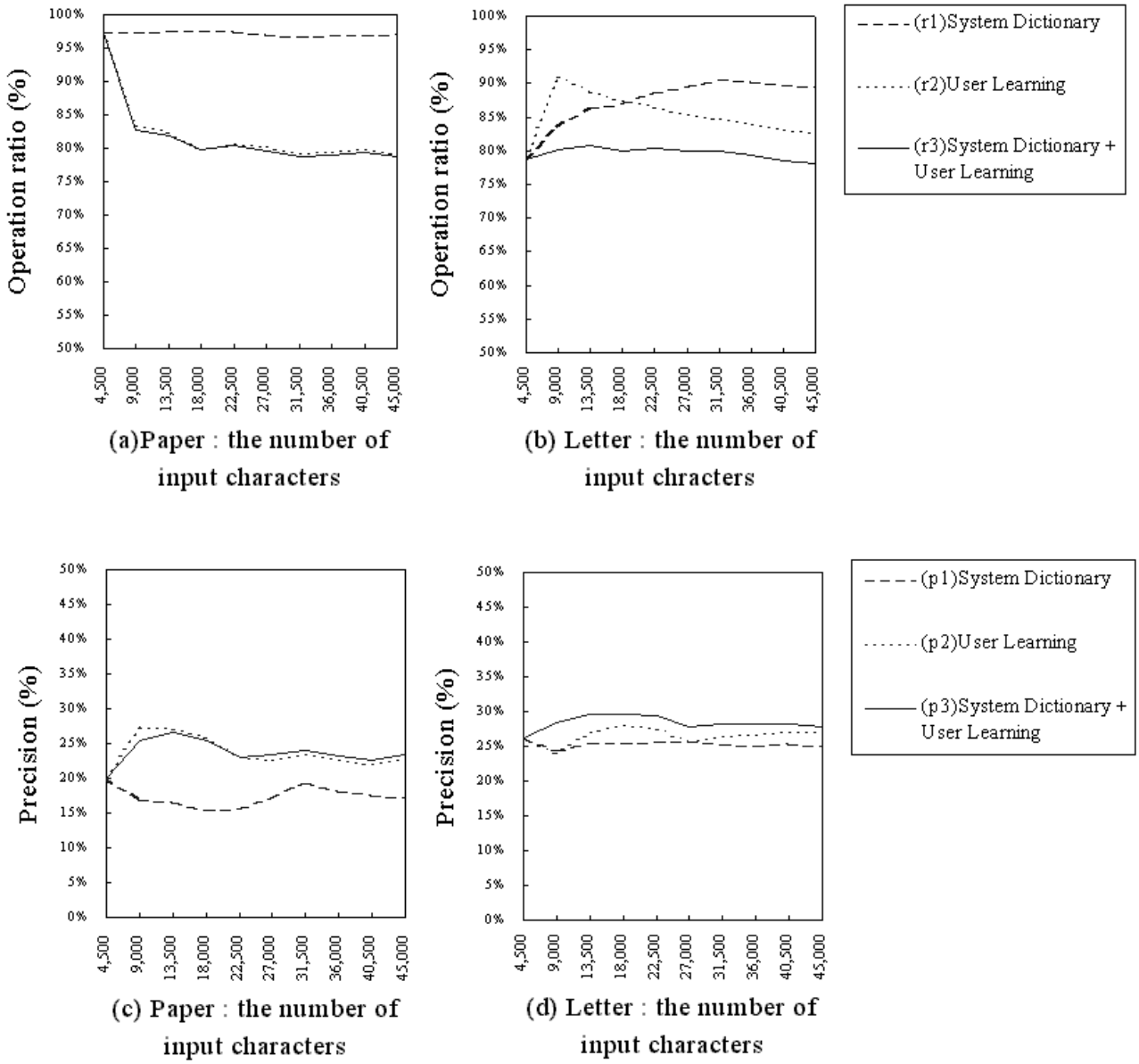

Figure 3: Experimental Results. (a)Operation ratio for the paper. (b)Operation ratio for the letter. (c)Precision for the paper. (d)Precision for the letter.

Fig.3(d)). When 45,000 *kana* characters were typed, an average of the operation ratio was 78%, that is, a 22% decrease in the original operations was obtained; an average of the precision was 25%, that is, a quarter of shown prediction menu windows contained appropriate choices. This precision was enough to realize comfortable operations.

Contribution of the system dictionary: Using only the system dictionary, for the paper, an operation ratio was 97.6–96.6% (line (r1) in Fig.3(a)), that is, a 2.4–3.4% decrease in the original operations was obtained; for the letter, an operation ratio was 90.6–78.8% (line (r1) in Fig.3(b)), that is, a 9.4–21.2% decrease in the original operations was obtained. As a result, the system dictionary is effective for a text like a letter with idioms or common phrases, because

the system dictionary includes a lot of such phrases. Furthermore, compared a precision using both the system dictionary and user learning with that using only user learning, the former was worse for the paper (lines (p2) and (p3) in Fig.3(c)(d)). As a result, for some kind of texts, the system dictionary not only is ineffective but also reduces a precision.

Contribution of user learning: User learning had an effect for an operation ratio after more than 9,000 *kana* characters were typed (lines (r2) in Fig.(a)(b)). In fact, if the user types about ten pages of texts, a 15–20% decrease in the original operations can be obtained.

5 Conclusions

We proposed a *kana-kanji* conversion system with input support based on prediction. Our system features:
(i) It shows prediction candidates of *kanji* character strings which the user intends to input without any special key operation. If no appropriate choice is presented, the candidates disappear automatically when the next *kana* character is entered.

(ii) Arbitrary positions of typed *kana* character strings are regarded as the top of words.

(iii) A system dictionary and a user dictionary are used, and each entry in the system dictionary has *certainty factor* calculated from the frequency of words in corpora.

(iv) Candidates are estimated by *certainty factor* and *usefulness factor*, and likely ones with greater factors than thresholds are shown.

(v) Words and phrases are extracted from typed texts, and registered into the user dictionary automatically.

The proposed system could reduce the user's key input operations to 78% from the original ones in the experiments using two kinds of texts. This system was built into the TOSHIBA Japanese word processor, the *JW-8020*, which was released in autumn 1998 (Fig.4).

To raise a precision by field information and context of texts is our future work.

Figure 4: TOSHIBA Japanese word processor, the *JW-8020*, where the proposed system was built.

References

Alice Carlberger, Johan Carlberger, Tina Magnuson, Sira E. Palazuelos-Cagigas, M. Sharon Hunnicutt, and Santiago Aguilera Navarro. 1997. Profet, a new generation of word prediction: An evaluation study. In *Proceedings of the ACL Workshop on Natural Language Processing for Communication Aids*, pages 23–28, July.

John J. Darragh, Ian H.Witten, and Mark L. James. 1980. The reactive keyboard: A predictive typing aid. *IEEE Computer*, 23(11):41–49, November.

Toshikazu Fukushima and Hiroshi Yamada. 1996. A predictive pen-based japanese text input method and its evaluation (in japanese). *Journal of Information Processing Society of Japan*, 37(1):23–30.

Nestor Garay-Vitoria and Julio G. Abascal. 1997. Word prediction for inflected languages. application to basque language. In *Proceedings of the ACL Workshop on Natural Language Processing for Communication Aids*, pages 29–35, July.

Toshiyuki Masui. 1998. An efficient text input method for pen-based computers. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'98)*, pages 328–335. Addison-Wesley, April.

Masakatsu Sugimoto. 1997. Input scheme for japanese text with shk keycard (in japanese). In *IPSJ SIGMBL Report No.1*, pages 1–6, May.