Lexical Search Approach for Character-String Recognition

M. Koga[1], R. Mine[1], H. Sako[1], and H. Fujisawa[1]

Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185, Japan
{koga,mine,sakou,fujisawa}@crl.hitachi.co.jp

Abstract. A novel method for recognizing character strings, based on a lexical search approach, is presented. In this method, a character string is recognized by searching for a sequence of segmented patterns that fits a string in a lexicon. A remarkable characteristic of this method is that character segmentation and character classification work as subfunctions of the search. The lexical search approach enables the parameters of character classifier to adapt to each segmented pattern. As a result, it improves the recognition accuracy by omitting useless candidates of character classification and by changing the criterion of rejection dynamically. Moreover, the processing time is drastically reduced by using minimum sets of categories for each segmented pattern. The validity of the developed method is shown by the experimental results using a lexicon including 44,700 character strings.

1 Introduction

The importance of optical character readers (OCRs), which are used for form processing, mail sorting, etc., is increasing. Especially, the technology to read sets of words such as names and addresses has become more important.

Many methods for recognizing words have been proposed. They are categorized into two groups: segmentation based methods and non-segmentation based methods. Non segmentation based methods such as HMM approaches or holistic approaches are powerful for words of Roman alphabets. However, they are not so successful for Chinese or Japanese characters so far in most cases. On the other hand, segmentation based methods perform rather good results for both.

In general, a process of a conventional segmentation based method consists of three stages: character segmentation, character classification, and linguistic processing as shown in Fig. 1. At each stage *a priori* knowledge, such as arrangement of characters, shape of characters, and sequence of characters, is used. Collaboration of these stages can improve the accuracy of recognition, because different types of *a priori* knowledge are available in the final decision. For example, the method presented in [3] uses character classification to select the best segmentation candidates. An example of collaboration of character classification and linguistic processing is presented in [6]. A method which uses the result of linguistic processing to select a sequence of patterns is presented in [2].

S.-W. Lee and Y. Nakano (Eds.): DAS'98, LNCS 1655, pp. 115–129, 1999.

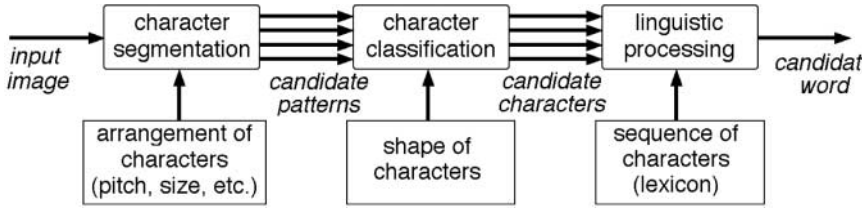

Fig. 1. Data flow model of the conventional word recognition methods.

The common strategy of these methods is the multiple hypothesis generation and testing. A stage sends multiple candidates to the following stages. The candidates are tested and the best one is selected in the following stages. Therefore, a lexicon is available only at the end of the whole process in this kind of methods.

In this paper, a new approach named "lexical search approach" is presented. It is such a top-down approach that the linguistic processing gives hypotheses of character categories to segmentation and classification. This approach has the potential to reduce the processing time and to improve recognition accuracy drastically. The approach is expected to be effective both for handwritten and machine printed strings. Similar ideas have been proposed [4][7] for word recognition of Roman alphabets. We applied the method to the recognition of names of places in Japan. We think this paper describes the first attempt to apply such an idea to a language in which a large number of characters are used. An improved technique to cope with the enormousness of characters is presented in the following sections. In addition, experimental results that show advantages of the method are presented.

2 Concept of Lexical Search Approach

We call such an approach "lexical search approach" that linguistic processing referring to lexicon controls character segmentation and character classification. As shown in Fig. 2, the data flow of the approach is completely different from that of the conventional approach. Linguistic processing gives the character segmentation information about the position where the next pattern may appear. Character segmentation extracts candidate patterns according to the information. Then character classification tests each segmented pattern immediately and decide whether it is acceptable contextually. Here the parameters of the character classifier is adapted dynamically to each pattern referring to the lexicon.

Accuracy and processing time of recognition can be improved by using the lexicon in the early stage of recognition. For example, suppose a lexicon includes the names of the counties in the world. In a conventional method, all candidate patterns should be classified into 26 categories (A to Z). However, if the first characters of a word is recognized as "J", then the following patterns should be classified only as "A" or "O", because "JAPAN", "JAMAICA", and "JORDAN" are the only words whose first characters are "J". The number of categories can

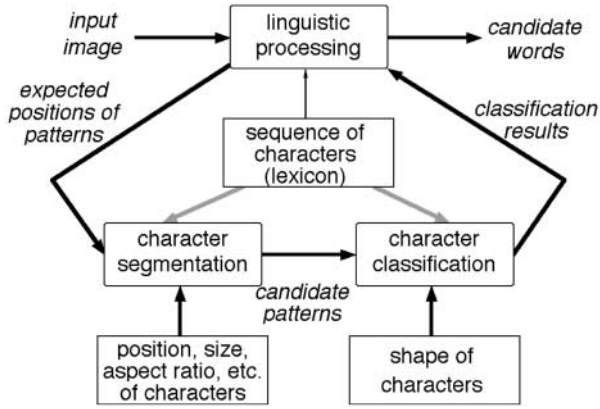

Fig. 2. Data flow models of the developed method.

be reduced from 26 to 2, and accuracy and speed can be improved. It is also possible to change the classification algorithm and parameters according to the set of possible categories for the following patterns.

Figure 3 illustrates the relationship among an input image, candidate patterns, and a lexicon. A name of a city in Japan, "川越市", is used as an example of an input image. The character-strings recognition is a process finding correspondence of patterns and characters in a word (as shown by gray arrows).

Some features of the proposed method are shown here. The candidate patterns are represented in terms of a graph named "segmentation hypothesis graph". Each arc represents a candidate pattern. Nodes represent boundaries of candidate patterns. As most Japanese characters consist of more than one connected components, many candidate patterns can be generated as shown.

Lexicon is represented as a Trie [1] named "lexicon tree" in our method, as illustrated in the figure. This kind of data structure is necessary to cope with the big lexicon where a large number of character categories are used. A path from the root to one of the leaves represents a word. In this example, "川口市", "川越市", "川崎市", "川西市", "川之江市" etc. are included in the lexicon.

3 Character-String Recognition Method Based on Lexical Search Approach

3.1 Overview

Our goal is recognition of phrases which represent names of places in Japan. Each of the phrases consists of a few words. Each of the words is a sequence of three or four Japanese characters in general. Because Japanese words are short, we devised a method to recognize a phrase as one character string.

The developed character-string recognition method has three sub-functions: pre-segmentation, word matching and selection of candidate words (Fig. 4). Pre-segmentation is a technique commonly used in the conventional segmentation

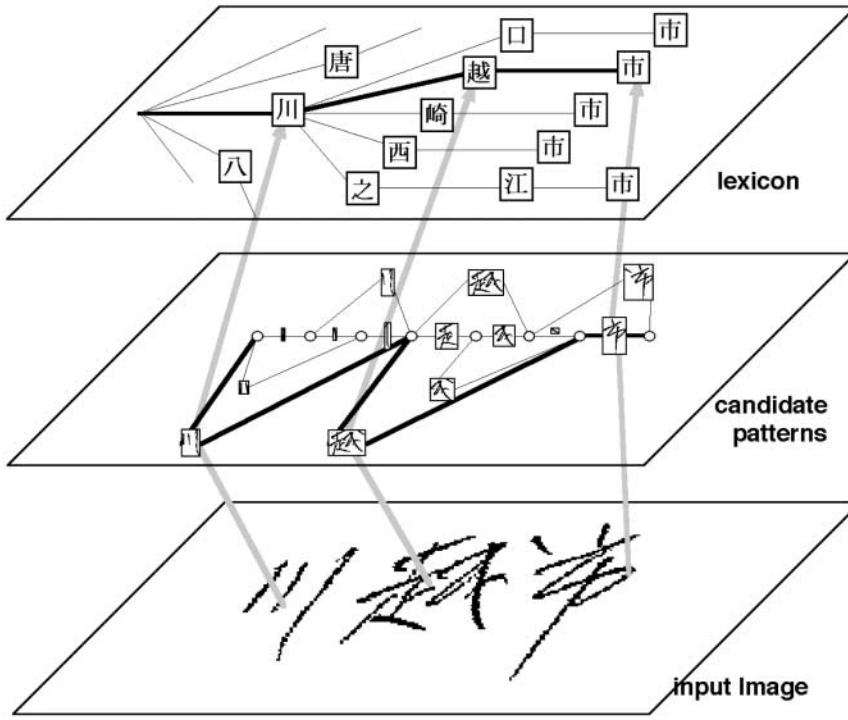

Fig. 3. Relationship among input image, candidate patterns and lexicon.

based methods [2][3]. In the pre-segmentation, a character string is cut into candidate patterns so that correctly segmented patterns are included in the candidates. Word matching is the body of this method. It is a process to search for a sequence of candidate patterns that fits a word in the lexicon. The selection of a candidate pattern is a sub-function of word matching. It selects candidate pattern(s) that can follow the previous one in the sequence. The character classification is also a sub-function of the word matching. It evaluates how candidate patterns fit characters in words. The word matching process outputs the result in terms of a search tree. Then the best candidate word is selected from the search tree.

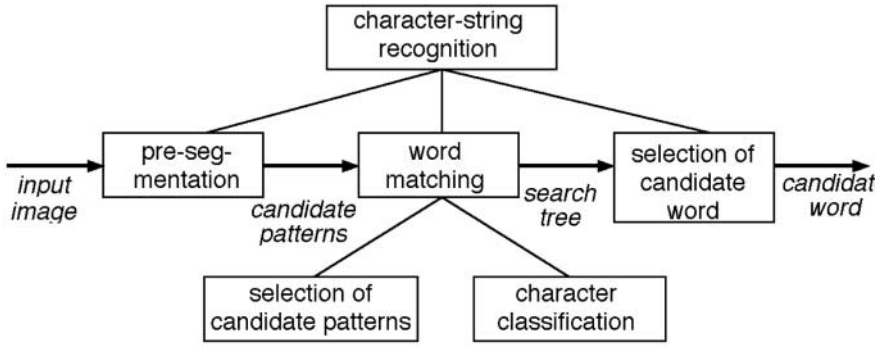

Fig. 4. Function tree of the developed method.

Note that the segmentation process is divided into the pre-segmentation and the selection of the candidate pattern. In the explanation of concept of the lexical search approach, the whole segmentation is done under the control of linguistic processing. However, to reduce processing time, we adopted the pre-segmentation method in this implementation.

3.2 Word Matching

There are many combinations of candidate patterns and characters in the lexicon. In the word matching, a simple breadth-first search algorithm is applied to search for a sequence of patterns that fits a word. A search tree generated from the example shown in Fig. 3 is illustrated in Fig. 5. Each node except the root node corresponds to a combination of a candidate pattern and a character in the lexicon. A recognized word is represented as a path from the root to one of the leaves. In this example, the word "川越市" is recognized.

A more detailed data structure of the tree equivalent to Fig. 5 is shown in Fig. 6(a). Each data record of the nodes has three pointers. The first one is the pointer to the upper node. If the upper node is the root, it points to NULL. The second one is the pointer to the node in the lexicon tree. The pointers A, B, C,

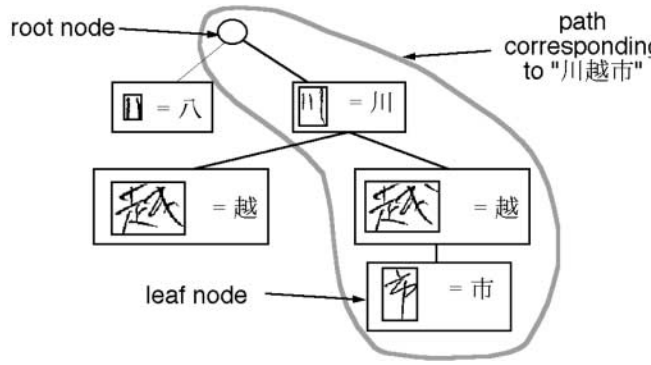

Fig. 5. Search tree. Each node corresponds to a combination of a candidate pattern and a character. A path from the root node to a leaf node corresponds to a recognized word.

and D store the addresses of nodes in the lexicon tree shown in Fig. 6(b). The third one is the pointer to the candidate pattern. The pointers a1, b1, c1, c2, and d1 store the addresses of the candidate pattern data (Fig. 6(c)).

An example of the search process is shown in Fig. 7. The process is divided into 5 steps ((a)-(e)). In the left box of each step, the search tree under processing is illustrated. The gray arrows indicate the focused nodes from which child nodes are being generated. In the middle box, the lexicon tree is illustrated. The gray arrows indicate the characters corresponding to the focused nodes. In the right box, the segmentation hypothesis graph is illustrated. The thick arcs indicate the candidate patterns that can follow the patterns corresponding to the focused nodes.

At the beginning, as shown in Fig. 7(a), the combinations of the first characters of each word and the candidate patterns are tested by character classification. If acceptable combinations are found, nodes for the combinations are generated. In this example, patterns that look like "八" and "川" respectively are found in the candidate patterns.

Next, the nodes generated in step (a) are focused on in (b) and (c). Combinations of following candidate patterns and following characters in the lexicon are tested. If acceptable combinations are found, child nodes for them are generated (c).

The descendant nodes are generated recursively in the same way as (d) and (e). If a node being focused on corresponds to a leaf in the lexicon tree, then the node is regarded as a leaf and no more child nodes are generated under it (e).

Note that it is not necessary to focus on only the leftmost patterns in the input image at the beginning of the search as shown in (a). If the search starts only from the leftmost patterns, the processing time can be reduced. However, OCR sometimes should find where the target word starts in the text line. In real applications, noise may be included in the text line by error of line segmentation. Unexpected words may exist besides words in the lexicon. The developed method

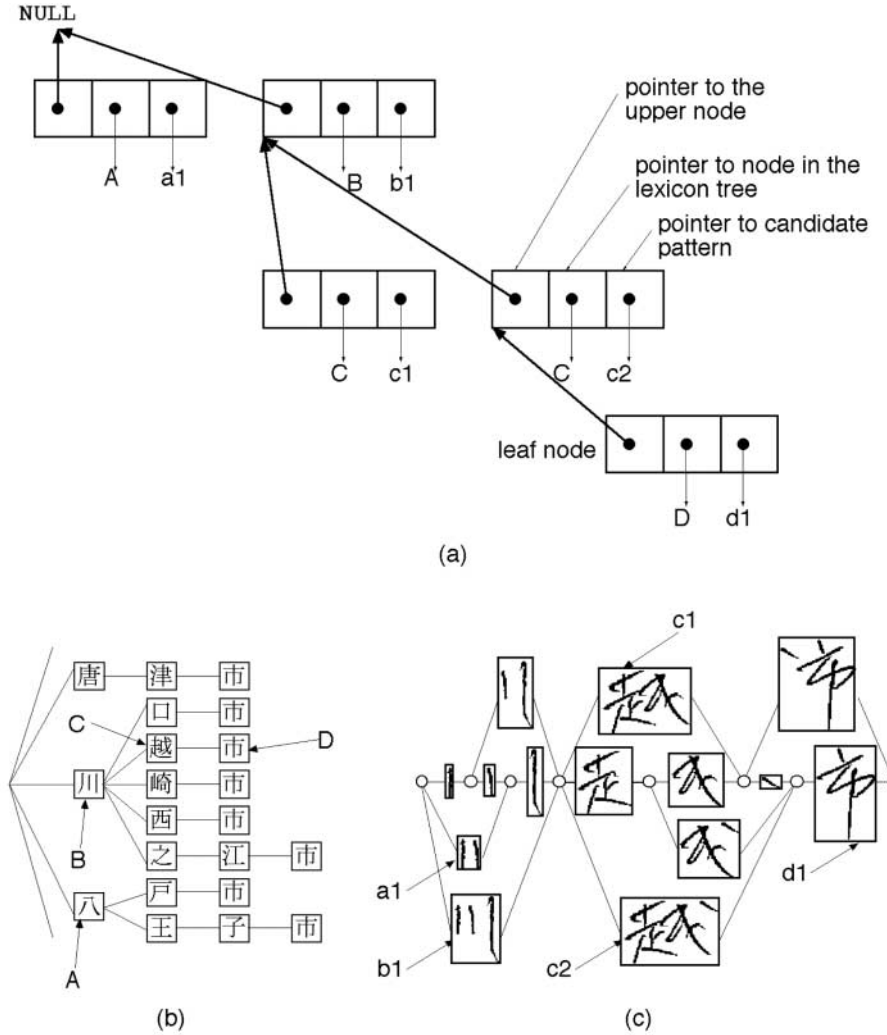

Fig. 6. Data structure of search tree. (a) Search tree. (b) Lexicon tree. (c) Segmentation hypothesis graph.

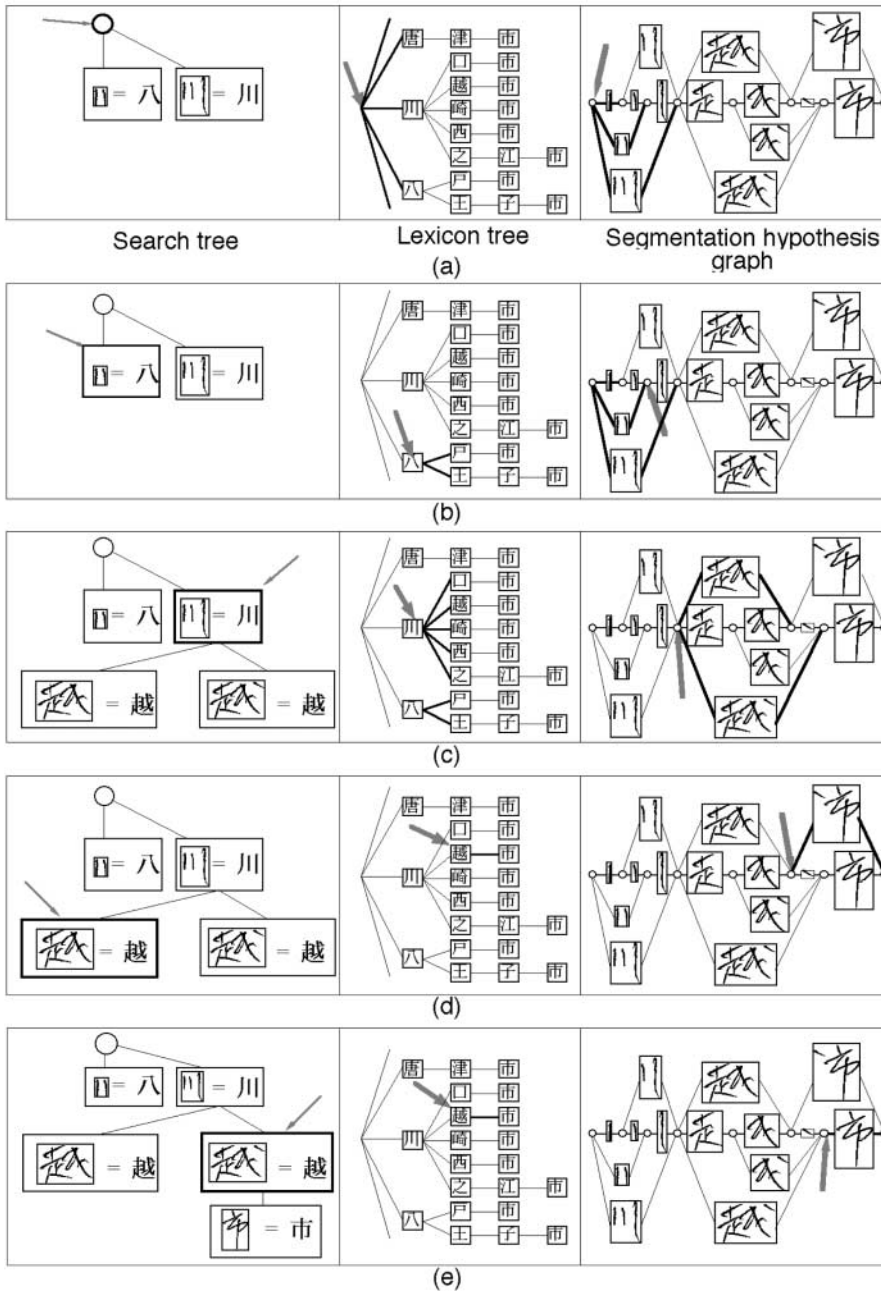

Fig. 7. Example of a search process.

can handle such cases only by checking all candidate patterns at the beginning of the search. Once the first characters of the words are found, the child nodes can be generated in the same way as explained above and the target words can be recognized.

3.3 Lexicon Tree

The data structure of the lexicon tree is illustrated in Fig. 8. It is a kind of Trie. A path from the root to a leaf represents a word in the lexicon. Each data record of a node in the lexicon tree consists of a character code, a pointer to the reference patterns and a pointer to the child node pointer table. The child node pointer table is a table of variable size that stores the pointers to the data records of child nodes. If a node is a leaf, the pointer to the child node pointer table points to NULL. The pointer to reference patterns point to the reference patterns used for the character classification. The reference patterns are used when the parent node is focused on in the word matching.

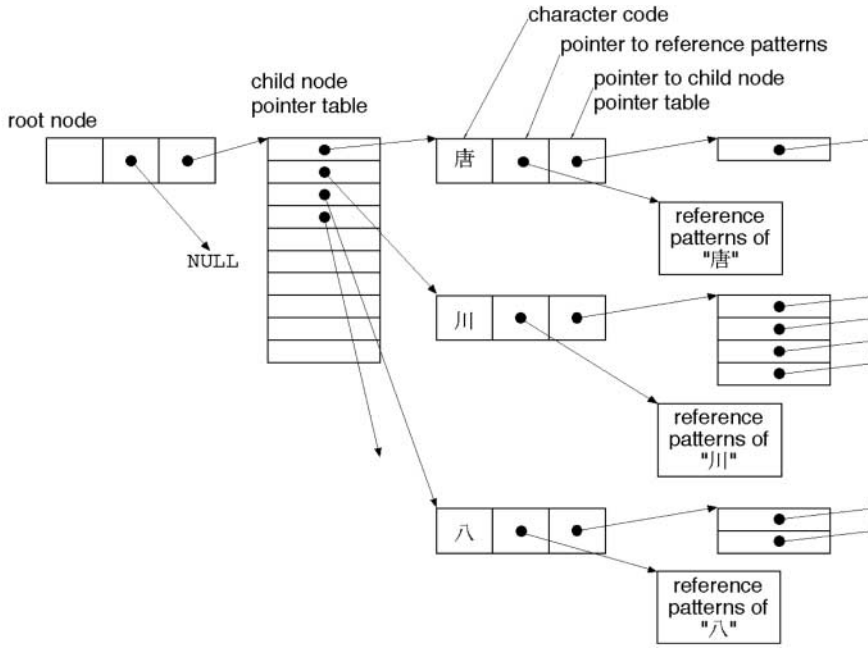

Fig. 8. Data structure of lexicon tree.

The style of the lexicon tree affects the processing time. A plain lexicon tree, which is equivalent to a list of words as in Fig. 9 (a), also can be used for the developed method. However, the nodes of the same characters and the same parent nodes can be merged as in (b) and the tree can be shrunk, and the

processing time can be reduced. If the tree is not shrunk, classification of the same pattern into the same set of categories is repeated, while such repetition is avoided with a shrunk lexicon tree. In the case that there are many words in the lexicon, the processing time using plain lexicon tree may be longer even than the conventional method .

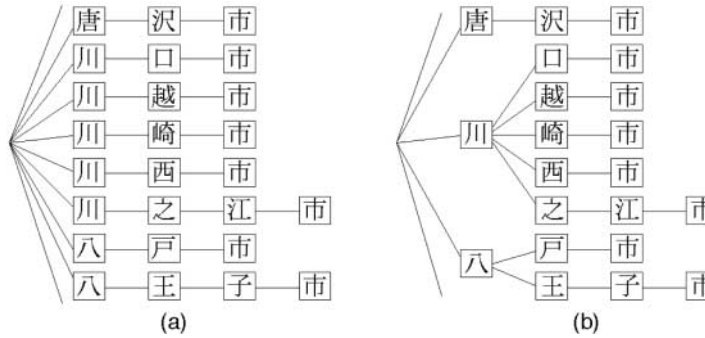

Fig. 9. Types of the lexicon trees. (a) Plain lexicon tree. (b) Shrunk lexicon tree.

A part of a real lexicon tree representing names of cities, counties, and towns in Japan is shown in Fig. 10.

3.4 Character Classification

A simple minimum distance method is used in the character classification in the developed method. The similarity is used instead of a distance. The feature vector represents the distribution of the contour direction of an input pattern. The categories of the reference patterns are selected referring the lexicon tree. For example, in the step (c) in Fig. 7, 5 categories ("口", "越", "崎", "西", and "之", are selected.

Special criteria for rejection are applied. Here, the two conditions to accept the i th category are shown below:

$$S_i > \delta \tag{1}$$

$$S_i \geq max\{S_i : 1 \geq j \geq N\} - \varepsilon \tag{2}$$

where
S_i : similarity of i th category
N : number of categories
δ : absolute threshold (constant)
ε : relative threshold (constant)

Condition (1) is used with the intention of selecting the categories whose reference patterns are similar enough to the input patterns. Condition (2) is used with the intention of avoiding the categories that give much smaller similarity

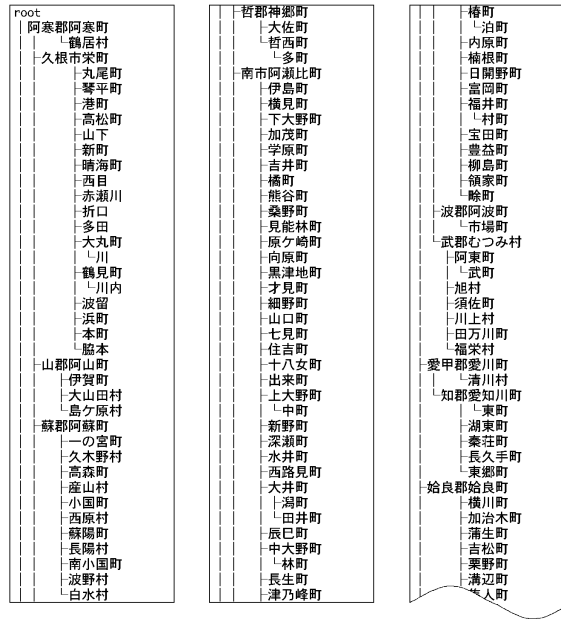

Fig. 10. Example of a lexicon tree that represents names of cities, counties, and towns in Japan. It consists of 44,700 phrases in total.

than others. When more than one categories fulfill the two conditions, all of the categories are accepted. Therefore, if a category gives higher similarity, the condition of rejection becomes severer. This is a good characteristic because a category with lower similarity may cause an error of final recognition result in such a case.

The two parameters, δ and ε , have a significant effect on the processing time and the accuracy of recognition. If δ is larger or ε is smaller, the processing time will become shorter and the error ratio of character string recognition will become lower. If δ is smaller or ε is larger, the acceptance ratio of total recognition will be higher.

The selection of reference patterns by the lexicon can increase the processing speed. More than 4,000 characters are used in the names of the places in Japan. To recognize them by conventional method, all candidate patterns should be classified into the all categories. On the other hand, less categories are needed in the developed method, because the number of characters each of which can appear as the first character of a city name or a county name is only 422. If the category of the first character is fixed, only one or two categories can follow it in most cases. The processing time of the developed method is thus estimated to be 1/100 of that of the conventional method in the best case.

3.5 Selection of Candidate Words

After the word matching process, the best candidate word is looked for in the search tree. First, the nodes that corresponds to leaves of the lexicon tree are selected. Then candidate words are found by tracing the search tree back using the pointers to upper nodes. Finally, the best candidate word is selected. In the developed method, a heuristic method is used for this selection. The word whose summation of similarity is the largest is selected as the best one.

4 Experimental Evaluation

4.1 Experimental Design

To evaluate the validity of the developed method, the accuracy, i.e. acceptance ratio and error ratio, and the processing time are tested. The acceptance ratio is the ratio of accepted images to the test images. If the character-string recognition program does not give any candidate words, the image is regarded as rejected; otherwise it is accepted. The error ratio is the ratio of errors to the accepted images. If the first candidate word is wrong, the result is regarded as an error.

The processing time of developed method is the time spent in the word matching including selection of patterns and character classification (see Fig. 4). The processing time of the pre-segmentation and the selection of candidate word is so small, comparing with the character classification, that it is ignored. A workstation with a 160 MHz RISC processor is used for the experiment.

Another character-string recognition program is used to compare the validity of the new method with the conventional method. The program uses the pre-segmentation program same as the new method. The same set of reference patterns is used in the character classification. The processing time of the classification is measured for the comparison to the new method. It classifies all candidate patterns into all categories in the lexicon. The classification is accelerated by a pre-classifier that roughly classify the input patterns before the main classification. Several sequence of candidate patterns are selected by dynamic programing [2] and then interpreted by post-processing [6].

4.2 Lexicon

The lexicon is designed to recognize the names of cities, counties, and towns in Japan. An expression of a Japanese address usually starts with a prefecture name followed by a city name or a county name, and then by the town name. Thus a set of combined words, each of which consists of a city name or a county name, and a town name, are used as the lexicon. If a combined word is longer than five characters, only five characters from the head are used.

The lexicon consists of 44,700 combined words. 1,915 kinds of characters appear in it. There are totally 76,167 nodes in the shrunk lexicon tree (as shown Table. 1).

Table 1. Numbers of nodes in the shrunk lexicon tree used in the experiment

layer Example	1	2	3	4	5
number of nodes	422	1,120	1,806	29,154	43,664

4.3 Test Images

As a set of test samples, 1,616 binary images of machine printed character strings (Figs. 11 and 12) are used. A test image consists of several line images. Line segmentation is done previously. A word in the lexicon is included somewhere in the segmented line images of each test image. Note that if the character-string recognition program find a false candidate word at a wrong position, it is regarded as an error.

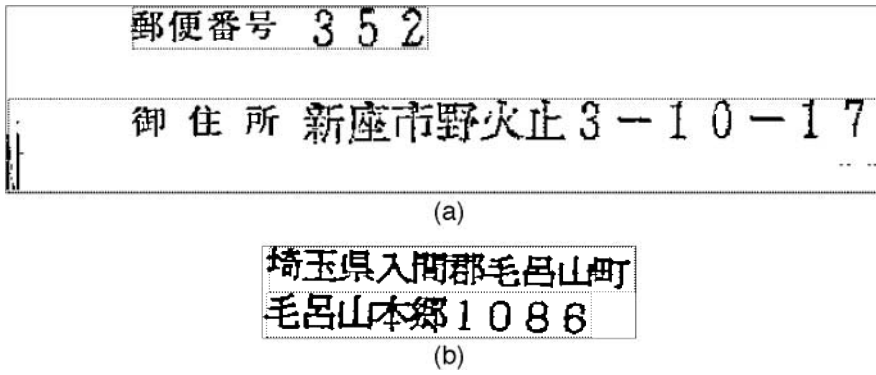

Fig. 11. Accepted test images. (δ = 0.63, ε= 0.05)

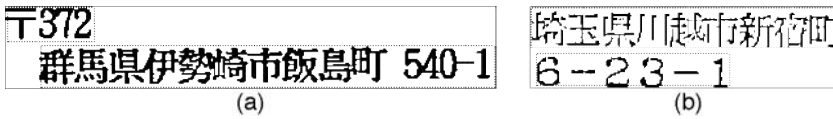

Fig. 12. Rejected test images (δ = 0.63, ε= 0.05)

4.4 Experimental Results

The processing times and accuracy of recognition of the developed method are listed in Table 2. The processing time of the developed method is much shorter

than that of the the conventional method. If the plain lexicon tree is used, the processing time is very long, as expected. If the search starts only at the leftmost patterns of the character strings, the processing time becomes much faster while the acceptance ratio becomes lower. The error ratio of the new method is smaller than the conventional method. Most of the errors of the conventional method come from reading wrong positions.

Table 2. Numbers of nodes in the shrunk lexicon tree used in the experiment

	conventional method	developed method		
		plain lexicon	shrunk lexicon	
	search starting at every position			search starting at the head of the text line
average processing time	0.74 sec	12.3 sec	0.12 sec	0.01 sec
acceptance ratio	72.4%	76.5%		32.6%
error ratio	2.8%	0.6%		0.6%

The correlations between the recognition accuracy and parameters of character classification is shown in Table. 3. As expected, the acceptance ratio increases when ε is large and δ is small.

Table 3. Accuracy of recognition. (acceptance ratio (%) / error ratio (%))

	δ=0.550	0.600	0.625	0.650	0.675	0.700	0.750
ε=0.01	85.7/ 9.7	81.9/1.5	78.1/0.3	73.4/0.2	67.0/0.3	57.7/0.2	32.1/1.2
0.02	87.1/10.9	82.5/1.4	78.9/0.4	74.2/0.3	67.7/0.4	58.2/0.3	32.3/1.2
0.03	88.1/12.0	83.1/1.7	79.6/0.5	74.9/0.3	68.2/0.4	58.7/0.3	32.5/1.2
0.05	87.8/12.9	83.3/2.2	79.9/0.5	75.3/0.3	68.5/0.5	58.9/0.3	32.6/1.1
0.07	86.9/13.9	83.4/2.5	79.8/0.6	75.2/0.4	68.5/0.4	59.1/0.4	32.6/1.1

Examples of test images are shown in Figs. 11 and 12. The results of line segmentation are shown by the doted rectangles. There is no target word in the first line of Fig. 11(a) and there are noises and a pre-print (that means "address") on the left of the target word. The name of prefecture is on the left of the target word in Fig. 11(b). The proposed method could find the target words even in such cases. Figure 12(a) is a rejected example because of failure of pre-segmentation. In this case, characters are touching at so many points that correct boundaries cannot be found. Figure 12(b) is a example of rejected test image because of low similarity. Some strokes are lacking because of the low quality of printing, so the similarity of the pattern is low.

5 Conclusion

The developed method based on the lexical search approach is completely different from conventional methods. In the method, character segmentation and character classification work as sub-functions of word matching. Similar ideas were recently proposed, but we believe this is the first attempt to apply the idea to a language in which a large number of characters are used. This method enables the character classifier to adapt the context, and it has a potential to improve processing speed and accuracy.

Experimental results show the validity of the method. The search using Trie improves the processing time. The mechanism to change the rejection criterion of character classification worked as expected. On the other hand, the method is sometimes not so robust, because the search process can not skip characters that give low similarity or misspelled characters. A simple breadth first search can not handle enormous combinations of patterns and characters if such skip is allowed.

The algorithm is simple and many kinds of extension are possible. Any kinds of character classifier can be adopted to this approach. A sophisticated search algorithm instead of a breadth-first search may improve the performance. Other data structures than the Trie can be used for the approach.

References

1. E. Fredkin, Trie Memory, Commun. ACM, Vol. 3, No. 9, pp. 490 - 500, (1960) 117
2. H. Murase, Segmentation and Recognition of Hand-Written Character String Using Linguistic Information, Trans. of the Institute of Electronics, Information and Communication Engineers, vol. J69-D No. 9, pp. 1292-1301 (1986 in Japanese) 115, 119, 126
3. H. Fujisawa et al, "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis, Proc. of the IEEE vol. 80, No. 7, (1992) 115, 119
4. F. Kimura et al, Improvements of a Lexicon Directed Algorithhm for Recognition of Unconstrained Handwritten Words, Proc. of ICDAR '93, pp.18-22(1993) 116
5. E. Cohen et al, Control Structure for Interpreting Handwritten Addresses, IEEE Trans. PAMI vol. 16. no. 10, pp. 1049-1056 (1994)
6. K. Marukawa et al, A Paper Form Processing System with an Error Correcting Function for Reading Handwritten String, Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 469-481 (1994) 115, 126
7. C. Chen, Lexicon-Driven Word Recognition, proc. of ICDAR '95, pp. 919-922 (1995) 116
8. J. J. Hull, Incorporating Language Syntax in Visual Text Recognition with a Statistical Model, IEEE Trans. PAMI vol. 18, no. 12, pp. 1251-1256, (1996)
9. R. G. Casey et al, A survey of Methods and Strategies in Character Segmentation, IEEE Trans. PAMI vol. 18, no. 7, pp. 690-706 (1996)
10. P. D. Gader et al, Neural and Fuzzy Methods in Handwriting Recognition, IEEE Computer, Feb. 1997, pp. 79-8 (1997)
11. L. S. Yaeger et al, Combining Neural Networks and Context-Driven Search for Online, Printed Handwriting Recognition in the NEWTON, AI Magazine, Spring 1998, 73-89 (1998)