Document Retrieval System Tolerant of Segmentation Errors of Document Images

Takeshi NAGASAKI†, Toshikazu TAKAHASHI†, and Katsumi MARUKAWA†
†Hitachi, Ltd., Central Research Laboratory
1-280 Higashi-Koigakubo, Kokubunji-shi, 185-8601
Tokyo, Japan
naga-t@crl.hitachi.co.jp, takahash@crl.hitachi.co.jp, marukawa@crl.hitachi.co.jp

Abstract

This paper describes a new document retrieval method that is tolerant of OCR segmentation errors in document images. To overcome the segmentation and recognition errors that most OCR-based retrieval systems suffer from, the proposed method consists of two processing phases. First, the OCR engine first generates multiple character-segmentation and recognition hypotheses. Then the retrieval engine extracts keywords from the recognition hypotheses by using lexicon-driven dynamic programming (DP) matching. We have applied this method to both handwritten and printed document images and have demonstrated its effectiveness in reducing false drops and false alarms.

1. Introduction

Information technologies have enabled the efficient production, transmission, and storage of digital documents. As a consequence, the amount of digital imaged documents is increasing at an accelerating rate in many business areas. We will not be able to take full advantage of this enormous store of document images without using new document image retrieval techniques, and. many investigators have already contributed to the development of these techniques [1,2,3].

Most document image retrieval systems are based on Optical Character Recognition (OCR). An embedded OCR engine is used to convert the document image into text codes, and then retrieval (e.g., keyword spotting) is performed on the OCR-ed text by using text-search techniques. Recognition and segmentation errors in OCR, however, limit the accuracy of text search and document retrieval tasks [4,5,6,7]. These OCR errors have many causes: touching characters, fragmented characters, the existence of non-character patterns, complicated arrangements, and so on. These situations occur frequently in Japanese documents where handwritten and printed documents are often intermixed. To develop reliable retrieval systems based on OCR, it is necessary to prevent the accuracy degradation due to segmentation and recognition errors.

Segmentation of handwritten characters is an essential problem in character recognition technology. Character segmentation is one of the most difficult and important subtasks of Japanese character recognition because most Japanese characters (Kanji's) are composed of several small separate sub-patterns that make it difficult to segment individual characters correctly. One of the segmentation methods effective for segment Japanese characters is over-segmentation [8,9], in which the input image is cut into many parts that are then combined hypothetically in such a way that true patterns are included in the combined patterns. It is well known that a feedback from the contextual analysis using the result of character recognition in the sequence analysis can improve the recognition accuracy in general [10,11,12].

Using lexicon knowledge for error-collection is a popular approach to solving the recognition and segmentation problem [10,12]., and Murakawa and colleagues explain how OCR results can be made more accurate by using an error-collection method based on lexicon-driven language analysis [11]. It is difficult, however, to construct lexicon dictionaries because they vary with the subjects of documents, and sometimes the lexicon needed to improve character recognition and segmentation errors is not available because there is no dictionary adapted to the subject in question. In this paper we describe a keyword/document retrieval method based on the OCR hypothesis that contains multiple candidates for text line, character recognition, and segmentation results. It is an extension of a conventional method that uses several candidates in OCR [4,13,14].

2. Retrieval system based on the OCR hypothesis

2.1. Structure of retrieval system

Figure 1 shows the block diagram of our retrieval system. It consists of two parts: an OCR processing

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

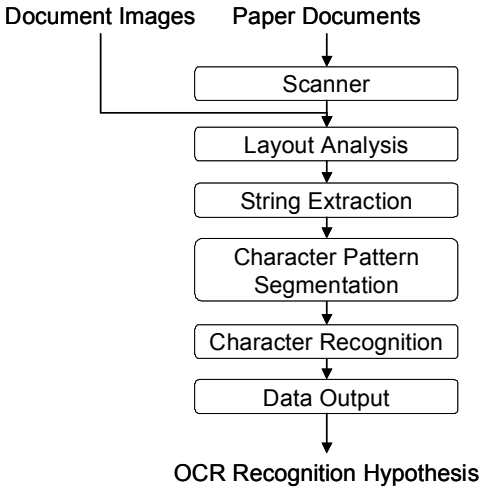
engine, and a keyword/document retrieval engine. The characteristics of this retrieval system are as follows:

1) The OCR engine generates multiple segmentation and recognition hypotheses to compensate the ambiguity of machine reading.

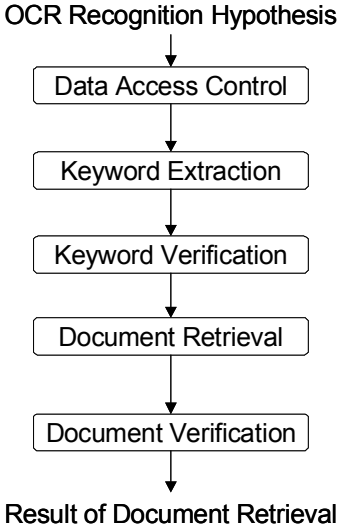
2) A subset of the regular grammar is used to represent the various keywords, which are the retrieval targets.

3) A dynamic programming (DP) matching algorithm is used to interpret the subset of regular-grammar keywords extracted from the OCR hypotheses by error-correcting matching.

4) When keyword candidates are being extracted by DP-matching, a keyword verification procedure is used to reduce the occurrence of false-alarm errors in retrieval (over-extracting) by computing the confidence of character recognition and pattern arrangements.



a) OCR engine



b) Retrieval engine

Fig. 1: Document retrieval system.

2.2. OCR hypothesis

The OCR hypothesis consists of several candidate text lines, segmented character patterns. and recognition results. The line hypothesis representing a text line includes information about the locations of character patterns, recognition candidates, recognition likelihood, and the relation of pattern sequence. This information can be modeled as a directed acyclic graph in which a node indicates a character segmentation point and an edge indicates a character pattern (Figure 2).

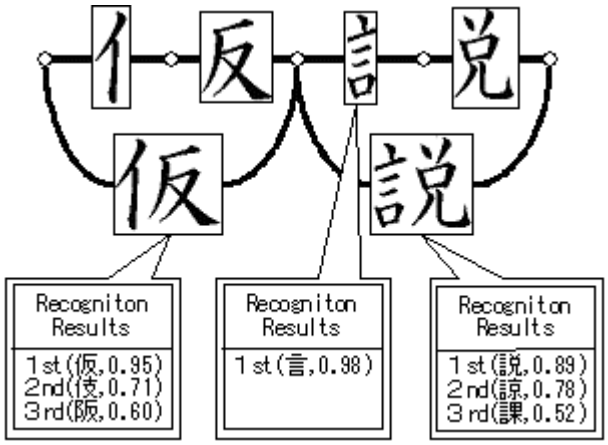


Fig. 2: OCR hypothesis (text line).

2.3. Keyword retrieval

The advantage of using the dynamic programming (DP) technique for string matching is that it can compute the best edit-distance metric and is tolerant disturbance such as character recognition error, noise insertion, and lack of character pattern. We used a DP algorithm to extract keywords from the OCR hypothesis, and we improved the algorithm to make it better able to handle the grammatical notation of various keywords.

RTN (Recursive Transition Network) grammar is often used to represent the ambiguity inherent in the definitions of words, but it is hard to interpret RTN by using the DP algorithm because RTN permits recursive definition of non-terminal symbols. In principle, the DP algorithm calculates a best- cost path on the directed acyclic graph. The matching problem in RTN grammar, on the other hand, becomes a directed cyclic graph problem because RTN permits recursive reference of symbols. We therefore use a subset of the regular grammar (SRG) for the notation of keywords. It should be emphasized that the level of SRG will affect the complexity of DP computation, so we will introduce several constraints on the SRG.

3. Lexicon-driven DP matching

3.1. Lexicon representation

The following grammar symbols are used in the SRG: the symbol "|" means arrangement of terms, the symbols "(" and ")" enclose a meaningful portion of terms, the symbols "[" and "]"enclose a terms that can be omitted. The symbols "*" and "?" are excluded from the SRG; they represents repetition of terms and matching with any characters. Figure 3 shows the definition of grammar symbols interpreted by the lexicon-driven DP matching algorithm.

The target keywords are represented with this SRG and we use three levels of representation to be considered with respect to the calculation complexity of the DP algorithm. The differences between these notations are related to the numbers of front (parent) symbols in the grammar sequence, and we consider Level-1 and Level-2 notations to develop lexicon-driven DP matching.

1) Level-1: Simple notation

The example of this notation is as follows. It includes four keywords – "ROBO", "ROBODOC", "ROBOT", and "ROBOTIC".

S (ROBO | ROBODOC | ROBOT | ROBOTIC) E

2) Level-2: Trie-structure notation

The trie-structure notation of upper keywords becomes to the following. The trie-form provides compact and short definitions for searching keywords.

S (ROBO (DOC | [T [IC]])) E

3) Level-3: Non-limited notation

This notation permits internal branching of words. The following example expresses the variation of a word "ROBO", which is as the result of misspelling or of someone mistaking the legendary wolf king "LOBO" for a robot.

S ((R | L) OBO (DOC | [T [IC]])) E

Symbols	Function
S	Start symbol of notation. Appears only once in the top of the definition.
|	Arrangement of terms.
(	Start of meaningful set of terms.
)	End of meaningful set of terms.
[	Start of omittable terms.
]	End of omittable terms.
E	End symbol of the notation. Generally, appears only once in the tail of the definition.

Fig. 3: Grammar symbols.

3.2. DP matching

This section introduces the DP equation interpreting the level-2 grammar. The OCR hypothesis and SRG are modeled as directed acyclic graph (DAG). We represent the OCR hypothesis of a text line as:

$$N = \{ (ns_i, ne_i, n_i) \mid i = 1, \cdots, |N| \}, \qquad (1)$$

and represent the SRG that defines the keywords to be extracted from document images as:

$$G = \{ (gs_i, ge_i, g_i) \mid i = 1, \cdots, |G| \}, \qquad (2)$$

where

n , g : graph edge on DAG,

ns , gs : start graph node of corresponding edge,

ne , ge : end graph node of corresponding edge.

The edge g corresponds to a grammar symbol and the edge n corresponds to a character pattern included in OCR hypothesis. We also introduce an edge set of a given edge e_j on the graph E, denoted as $\mathrm{PreE}(E, e_j)$. The edge set $\mathrm{PreE}(E, e_j)$ called front edge set of edge e_j, consists of edges that connect to the start node of e_j on the DAG E. More precisely, $\mathrm{PreE}(E, e_j)$ = $\{ e_i \mid ne_i = ns_j \}$. In terms of grammar, the front edge set corresponds to the grammar symbols placed in front of the symbol g_j. In terms of the OCR hypothesis, the front edge set corresponds to the character patterns placed in front of the character pattern n_q, denoted as $n_p \in \mathrm{PreE}(N, n_q)$.

The DP process can be considered as a minimum cost path search strategy. On the basis of upper notation, the matching cost $Cost(g_j, n_q)$ between a grammar symbol g_j and a character pattern n_q equals the following:

$$Cost(g_j, n_q) = \min_{\substack{g_i \in \mathrm{PreE}(G, g_j) \\ n_p \in \mathrm{PreE}(N, n_q)}} \{ F(n_p, g_i, n_q, g_j) \} \qquad (3)$$

$$F = Cost(g_i, n_p) + Path(n_p, n_q) + Next(g_i, g_j) + Match(g_j, n_q) \qquad (4)$$

where

$Match(\cdot)$: Cost of a single pattern-matching between grammar symbol g and character pattern n, which gives a penalty if the symbol is not one of the recognition candidates,

$Path(\cdot)$: Cost of the connection between character pattern n_p and n_q, which gives a penalty if the two patterns are not in the same neighborhood,

$Next(\cdot)$: Cost of the connection cost between grammar symbols g_i and g_j, which gives a penalty if the two symbols are not sequential.

The minimum cost of the path from the start to the end is obtained by computing the above functions for all g .. Given that $|G|$ is the length of grammar and $|N|$ is the number of the character patterns, the computation complexity of this DP process is of the order $|G||N|^2$ in the worst cases. But it could be reduced to the order $|G||N|$ by pruning the connections of character patterns.

3.3. Stack operation

The process of DP matching can be calculated with the stack table. The grammar consists of several specific symbols S, E, (,), [,], | and character codes C. C is a terminal symbol of grammar; in short, it's a normal character. The DP algorithm sweeps the sequence of grammar symbols from left to right, and process the

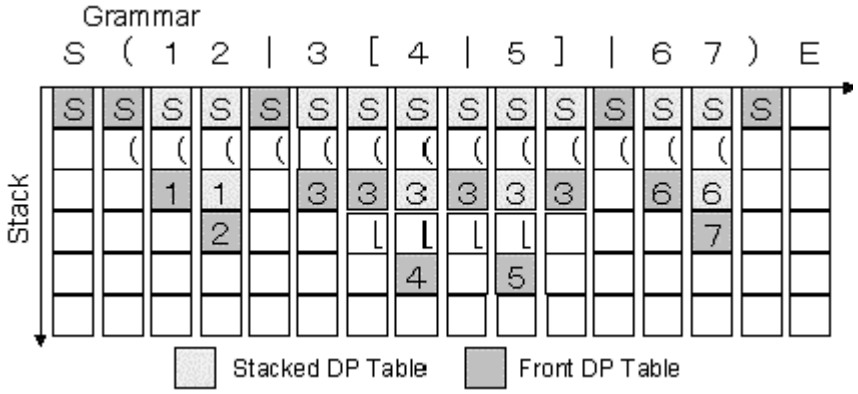


Fig. 4: DP table stack.

Symbol	Stack Operation
S	Initialize DP table.Set the last DPT as front DPT.
(	Push left the symbol to the stack.
)	This means the end of a word. Store computation results, the best path and cost of the word. Pop the stack until "(" is removed. Set the last DPT as front DPT.
[	Push the symbol "[" to the stack.
]	This means the end of a word. Store computation results. Pop the stack until "[" is removed. Set the last DPT as front DPT.
|	This means the end of a word. Store computation results. Pop the stack until "(" or "[" is coming. Set the last DPT as front DPT.
E	The end of computation.
C	Compute current DPT using front DPT. Push the current DPT on the stack. Set the lastest DPT on the stack as front DPT.

Fig. 5: Stack operation.

stack operations, push or pop the DP table (DPT) to the stack, at the each step of the sequence. The DP table is a container of best costs when a grammar symbol g_j matches with every character pattern $n_q \in N$.

For a given grammar symbol g_j, the DP table defined as like this:

$$Dpt(g_j) = \{(n_q, Cost(g_j, n_q)) \mid n_q \in N\}. \quad (5)$$

The best cost for the current target symbol g_j is computed from the previously calculated DP tables $Dpt(g_i)$, where the symbol g_j is one of the front symbols of g_j, is denoted as $g_i \in \mathrm{PreE}(G, g_j)$. When the grammar is level-2, it becomes $g_i = \mathrm{PreE}(G, g_j)$ since the number of front symbols becomes at most 1. The previously calculated DP table (front DP table) $Dpt(g_i)$ can be accessed in the DP table stack. Figure 4 shows state transition of the stack with the processing of DP computation. In this figure, the grammar defines a set of words:

{ 12, 3, 34, 35, 67 }

and its notation is presented in trie-form:

"S (12 | 3 [4 | 5] | 67) E".

Figure 5 shows stack operation for each of the grammar symbols.

4. Verification

The OCR hypothesis and lexicon-driven DP matching help keep the recall rate high because both approaches contribute to making keyword retrieval tolerant of recognition and segmentation errors in OCR. On the other hand, it increases the number of irrelevant keywords extracted from document images. To cope with this retrieval error, we utilize the peripheral features such as gaps between patterns, widths and heights of patterns, and evaluate the likelihood of extracted keywords (Figure 6). The likelihood of an extracted keyword is calculated by using the Bayesian rule according to the empirical distribution for each of the features of pattern arrangement.

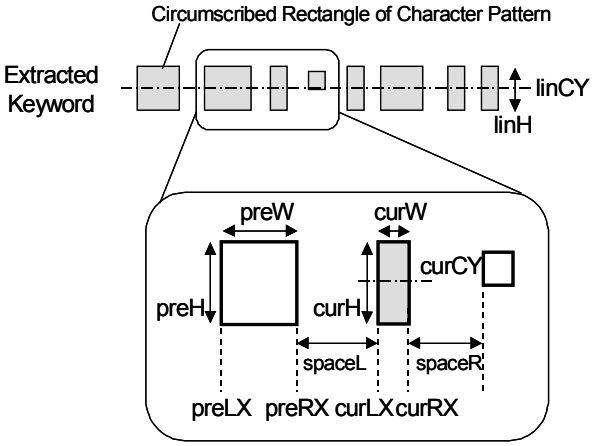


Fig. 6: Keyword verification.

5. Experiments

5.1. Database and criteria

We evaluated the performance of the proposed approach by carrying out experiments in keyword retrieval with the conventional retrieval method and the proposed one. these experiments used 400 sets of medical insurance documents, in which handwritten and printed texts are intermixed. Document images were scanned with 200dpi resolution and binary color. The documents contained 3182 text lines and 45091 characters. These documents were first encoded into text manually and relevant keywords were extracted using a conventional text search method. For retrieval keywords we select 550 keywords from a medical dictionary.

Table 1 shows the definitions of the performance criteria we used to evaluate the retrieval methods, and the retrieval accuracies obtained with the conventional method and the proposed method are listed in Table 2.

Table 1: Retrieval evaluation metrics.

```
recall    = B / A
precision= B / (B+C)
F metric = 2B / (A+B+C)
```

where,

A: the number of relevant keywords in documents,

B: the number of relevant keywords retrieved, and

C: the number of irrelevant keywords.

5.2. Experimental results

The retrieval accuracies obtained with several keyword extraction methods are listed in Table 2. The best recall rate, 97.8%, was obtained by using recognition hypotheses and the lexicon-driven DP algorithm. The precision, however, was low because incorrect keywords were extracted from images. This problem was overcome by verifying pattern arrangements and the recognition confidence of extracted keywords. The proposed method was the most accurate, having an F metric of 0.9.

Table 2: Retrieval accuracies.

Retrieval methods	Recall	Precision	F metric
Traditional OCR and text searching.	73.50%	98.60%	0.84
Traditional OCR with multi-candidates of recognition results.	79.00%	95.80%	0.87
OCR hypothesis and lexicon driven DP.	97.80%	8.90%	0.15
OCR hypothesis and lexicon driven DP with verification.	90.30%	90.00%	0.9

6. Conclusion

This paper described a document retrieval method based on the OCR hypothesis and lexicon-driven DP matching. We evaluated its utility experimentally in searches of documents containing both printed and handwritten texts. The proposed method achieved 97.8% recall rate in maximum and an F metric of 0.9 respectively, which was about 17pt higher than that of conventional method. This result shows the advantage of this method in keyword retrieval and document retrieval tasks.

References

[1] J.L. DeCurtins and E.C. Chen, "Keyword spotting via word shape recognition," Proc. of SPIE, pp. 270–277 (1995).

[2] T. Kameshiro, T. Hirano, Y. Okada, and F. Yoda, "A document retrieval method from handwritten characters based on OCR and character shape information," Proc.ICDAR'2001, pp. 597–601 (2001).

[3] K. Kise, Y. Wuotang, and K. Matsumoto, "Document image retrieval based on 2D density distributions of terms with pseudo relevance feedback," Proc of 7th ICDAR, pp. 488–92 (2003).

[4] K. Marukawa, H. Tao, H. Fujisawa, and Y. Shima, "Document retrieval tolerating character recognition errors - evaluation and application," Pattern Recognition, vol. 30, pp. 1361–1371 (1997).

[5] S. Senda, M. Minoh, and K. Ikeda, "Document image retrieval system using character candidates generated by character recognition process," Proc. of 2nd ICDAR, pp. 541–546 (1993).

[6] D. Lopresti and J. Zhou, "Retrieval strategies for noisy text," Proceedings of Fifth Annual Symposium on Document Analysis and Information Retrieval, pp. 255–269 (1996).

[7] Y.H. Tseng and D.W. Oard, "Document image retrieval techniques for Chinese," Proceeding of Symposium on Document Image Understanding Technology, pp. 151–158 (2001).

[8] H. Murase, "Using linguistic information for segmentation and recognition of hand-written character strings" (in Japanese), Trans. of the Institute of Electronics, Information and Communication Engineers, vol. J69-D, no. 9, pp. 1292–1301 (1986).

[9] H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation methods for character recognition: From segmentation to document structure analysis," Proc. of the IEEE vol. 80, no. 7, pp. 1079–1092 (1992).

[10] D.A. Dahl, L.M. Norton, and S.L. Tailor, "Improving OCR accuracy with linguistic knowledge," Proc. of 2nd SDAIR, pp. 169–177 (1993).

[11] K. Marukawa, K. Nakashima, M. Koga, Y. Shima, and H. Fujisawa, "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).

[12] G. Kim and V. Govindaraju "A lexicon-driven approach to handwritten word recognition for Real-Time Application," IEEE Trans. PAMI, vol. 19, no. 4, pp. 366–379 (1997).

[13] H. Fujisawa and K. Marukawa, "Full-text search and document recognition of Japanese text," Proc. of 4th DA&IR, pp. 55–80 (1995).

[14] Y. Katsuyama, H. Takebe, K. Kurokawa, "Highly accurate retrieval of Japanese document images through a combination of morphological analysis and OCR," Proc. of SPIE, Document Recognition and Retrieval IX, vol.4670, pp.57-67 (2002).