Development of Template-free Form Recognition System

Junichi Hirayama, Hiroshi Shinjo, Toshikazu Takahashi, and Takeshi Nagasaki
Central Research Laboratory, Hitachi, Ltd.
Kokubunji, Tokyo 185-8601, Japan
junichi.hirayama.qq@hitachi.com

***Abstract*—We present a new form recognition technique. In our work, we were especially interested in developing a "template-free" form recognition technique that extracts and recognizes target characters without pre-defined layout knowledge (form-template). We also attempted to overcome well-known difficulties in developing template-free form recognition techniques, i.e., extracting items from noisy form images and ambiguous alignment layout forms. We were able to use a hypothesis testing approach to successfully extract such items from such form images.**

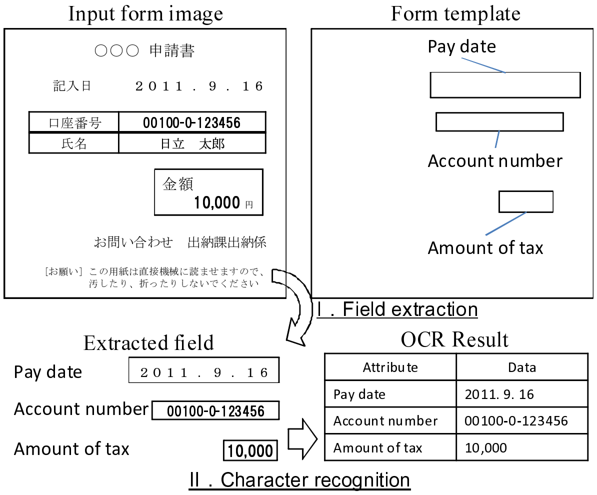
***Keywords*-Form Recognition, Meta Extraction, Document Layout Analysis, Character Recognition**

I. Introduction

A great many documents, or forms, are used in business every day. In government offices and banks, for instance, personnel have to handle forms that clients use to pay taxes and pay for utilities such as gas and water. Attempts have been made to make their work load easier by imaging standard forms from which they can acquire data. A traditional way of doing this, one that has been used by many businesses, is to use optical character recognition (OCR) in the handling of forms. Using OCR has improved work quality in these businesses and greatly reduced the amount of time that personnel spend in handling forms. In most cases where OCR is used, templates are formed to enable users to know where target strings are printed. These may include items such as the due date, the amount of money, and the client account number. However, there are many forms for which the formation of such templates cannot be applied. This is because each form layout needs its own form template and asking the workforce members to make a new form template on their own is quite a costly task.

To resolve these issues, we present a method of extracting and recognizing target characters without using a pre-defined form template. We call this method the "Template-free Form Recognition" technique.

The rest of the paper is organized as follows: In Section Ⅱ, we introduce the concept of template-free form recognition and problems we needed to solve. In Section Ⅲ, we present our proposed template-free form recognition algorithm. We evaluate our method in Section Ⅳ and summarize with a review of key points in Section Ⅴ.



Attribute	Data
Pay date	2011. 9. 16
Account number	00100-0-123456
Amount of tax	10,000

Figure 1. Form OCR flow

II. Outline of Form Recognition System

A. Form Recognition based on Form-template

One of the major business uses of form OCR is reading strings on form images that have attributes such as the date, the amount of money, and various payment sheet numbers. When we use such OCR products, we usually have to predefine a coordinate and a target string attribute. We also need to make a form template that has a correlation with the target form layout.

Figure.1 shows an example recognition flow on form OCR. Most form OCR products mainly consist of two major modules, i.e., a field extractor and a character recognizer. The field extractor first refers to the form template to extract the target character areas (e.g., the date, account number, and amount of payment). Here, the form template has a position and an attribute of target strings. The character recognizer then uses the field extractor to read the extracted area and outputs the recognition results and their attributes.

Such "template-based form OCR" is applicable to businesses that handle only single-formed or particular application-dependent layout forms (Where all forms have the same fixed layout). On the other hand, when there is need to handle various layout forms at one time, such as in the case of batch processing, template-based form OCR

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.56

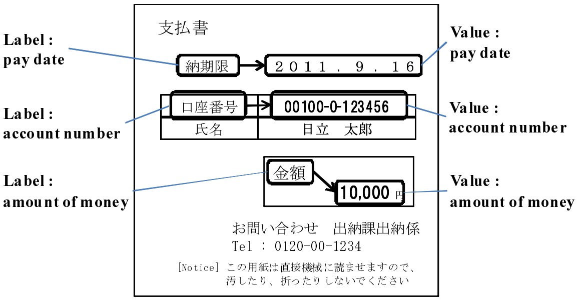


Figure 2. Relation between ”label” and ”value

cannot be used to recognize or analyze different layout forms. This is because each form layout needs to have its own correlating form template and form templates are made manually. Thus, it is quite a costly task to define a new form template. Hence, when the number of form layouts becomes larger, it becomes almost impossible to make form templates manually. Therefore, it is essential to develop a new form recognition method that can extract and recognize target characters without pre-defined form templates, i.e., “Template-free Form Recognition”.

B. Concept of Template-free Form Recognition

The basic concept of template-free form recognition is the extracting and recognizing of label-value relations from all strings in a form image without pre-defined layout knowledge. Here, ”label” and ”value” are item components. “Label” is a keyword that indicates a data attribute and a data heading and “value” is a string that means data. Label and value examples are illustrated in Figure 2. Extracting the relation between ”label” and ”value” allows us to determine where target strings are printed and what the target strings mean.

C. Related Work

Over the past few years, a number of studies have been made on relation extraction between two strings without pre-defined layout knowledge [2], [3], [4], [5], [6].

In [3], a method for analyzing the specific logical structure of a table layout was reported. In this method, a logical structure template was described by a Bayesian framework as an isomorphic graphical model of the generic logical structure and is pre-defined by users. The template is also used in attempts to match the logical target form layout structure by belief propagation. It was demonstrated that his approach is effective for three form types having an application-dependent unknown layout. To cite other cases, meta-data extraction from a portable document format (PDF) document was proposed in [4], [6]. Methods of this type may indicate desire on the part of the authors to extract certain paper characteristics such as title, author, author affiliation, figure captions, and chapter indexes. Most PDF analysis approaches commonly rely upon physical features of the document. Physical features such as zoning, font size, font style, and geometric information are obtained from digitized PDF information. However, these features do not include noises or recognition errors.

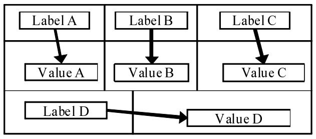


Figure 3. Orderly table layout

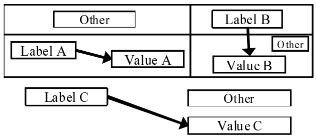


Figure 4. Non-orderly cell layout

Previously, we introduced a label-value extraction method in [2]. In this method, label-value relations can be extracted by referring to a pre-defined label list and by checking cell-linking from an orderly table layout form. This procedure is as follows. First, label strings are extracted from all recognized strings by matching to the label list. The label list is a list of label strings and their attributes that are common to all form layouts. Values are then extracted from cells adjacent to label cells.

D. Tasks

As mentioned in the above section, several label-value extraction methods have already been reported. However, little attention has been given to develop a method that solves the following problems simultaneously.

- Extracting label strings and value strings from noisy/low quality documents
- Extracting label-value relation alignment relations from ambiguous layout forms

The details of these problems are described as follows.

1) Relation extraction from noisy image: It is quite difficult to maintain high accuracy in character recognition efficiency for noisy/low quality images. Label and value strings are extracted by matching their recognition results to pre-defined label and value keywords. Thus, if the recognition results include some mis-recognized character results, the label-value extraction results should fail. Therefore, the mis-extraction of the label-value relation increases for noisy images.

2) Relation extraction from ambiguous layouts: Most early researches focused on analyzing an orderly table layout based on a cell-linking relation (Figure 3). On the other hand, in the case of analyzing a non-orderly cell layout, it is insufficient to check only cell-linking information for determining a positional relation between two strings. Since the alignment pattern of two strings not only has an adjacent cell relation but also one cell relation and a neighboring string relation, the positional relation may contain ambiguous alignment (Figure 4).

Our goal is to develop a label-value relation extraction method that can solve these problems simultaneously.

<Date> ::= “記入日” (W1) | “作成日” (W2) | “ご利用日” (W3) ;

<Number> ::= “会員番号” (W4) | “登録番号” (W5) ;

<Amount> ::= “入金額” (W6) | “金額” (W7) | “納付額” (W8) ;

Figure 6. Label list

<n> ::= “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” | “0” ;

<Date> ::= <n><n> “.” <n><n> “.” <n><n> |
<n><n> “/” <n><n> “/” <n><n> ; (D1)

<Number> ::= <n><n><n> “-” <n> “-” <n><n><n> ; (D2)

<Amount> ::= “¥” <n><n><n><n> | “¥” <n><n><n><n><n> |
<n> “,” <n><n><n> | <n><n> “,” <n><n><n> ; (D3)

Figure 7. Value dictionary

III. Proposed Method

A. Concept of Proposed Method

In our method, label-value relations are extracted by calculating three different score types, i.e., label score, value score, and alignment score. These scores are calculated independently by each module. Label-value relations are determined by integrating these scores. In other words, to test all possible label-value relation candidates exhaustively, our method generates each hypothesis independently and finally verifies all combinations of these generated hypotheses. Hence, our method cannot possibly maintain robustness for ambiguity vs. layout and character recognition errors.

An extraction flow is illustrated in Figure 5. Cells and strings are first extracted from a form image (step I). One cell and string extraction method has already been reported in [1].

Second, extracted strings are recognized (step Ⅱ). Recognition results are represented as a candidate lattice.

Next, three types of hypotheses are generated. Label score (step Ⅲ-a), Value score (step Ⅲ-b), and Alignment score (step Ⅲ-c) are calculated by each module independently. We describe these score calculation methods in the following sections.

Finally, label-value relations are determined by testing all the combinations of generated hypotheses. To be specific, this method extracts label-value relations based on link scores, which can be obtained by integrating label, value, and alignment scores (step Ⅳ,Ⅴ). We next describe in detail how to integrate calculated label, value, and alignment scores.

B. Label and value scores

Label score $Slabel(S_i, W_n)$ is the probability that a certain string S_i matches the Label W_n defined by Label list. The value score $Svalue(S_j, D_m)$ is the probability that a certain string S_j matches a syntactic rule D_m defined by the value dictionary. Label list and value dictionary scores are shown in Fig. 6 and 7, respectively. In both dictionaries, the BNF (Backus-Naur Form) apparently holds. In this case, labels W_1, W_2, and W_3 and the value's rule D_1 have the attribute of “date”, i.e., W_4, W_5, and D_2 have the attribute of “account number” W_6, W_7, W_8, and D_3 has the attribute of “amount of money”, respectively.

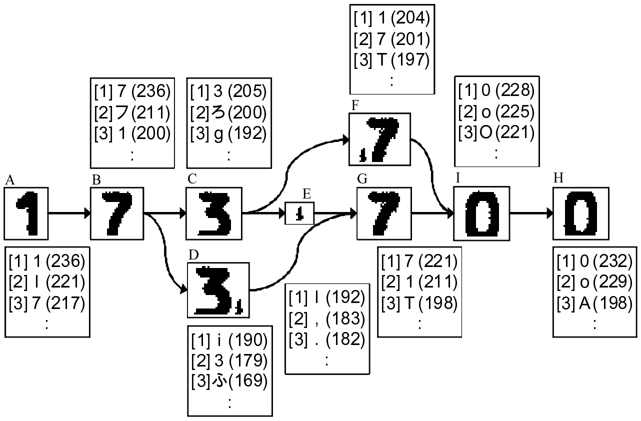


Figure 8. Candidate lattice

Each string recognition result is represented by the candidate lattice shown in Fig. 8. Each lattice node shows a segmented character image and its recognition candidates and scores.

Label and value scores can be calculated based on each individual character classifier scores of the candidate lattice. We defined label and value scores as being the average of the individual character scores as shown in (1),(2).

$$Slabel(S_i, W_n) = \frac{1}{C}\sum_{c=1}^{C} Clabel(S_i, W_n)_c \qquad (1)$$

$$Svalue(S_j, D_m) = \frac{1}{C}\sum_{c=1}^{C} Cvalue(S_j, D_m)_c \qquad (2)$$

where $Clabel(S_i, W_n)_c$ and $Cvalue(S_j, D_m)$ denote the classifier score of the c-th character (if S_i matches W_n and S_j matches D_m), respectively.

Figure 8 indicates an example of how to calculate value score. Referring to the syntactic rules of each attribute, dynamic programming searches the optimum partial path for each rule. In the case of Fig. 8, two syntactic paths are found. Path $A \to B \to C \to E \to G \to I \to H$ matches the amount of money attribute “173,700”(173000 yen) and path $A \to B \to C \to E \to G$ matches the date attribute “11.3.7”(mar. 7, 2011).

C. Alignment score

Alignment score $Salign(S_i, S_j)$ is the probability that a positional relation between two strings (S_i, S_j) is appropri-

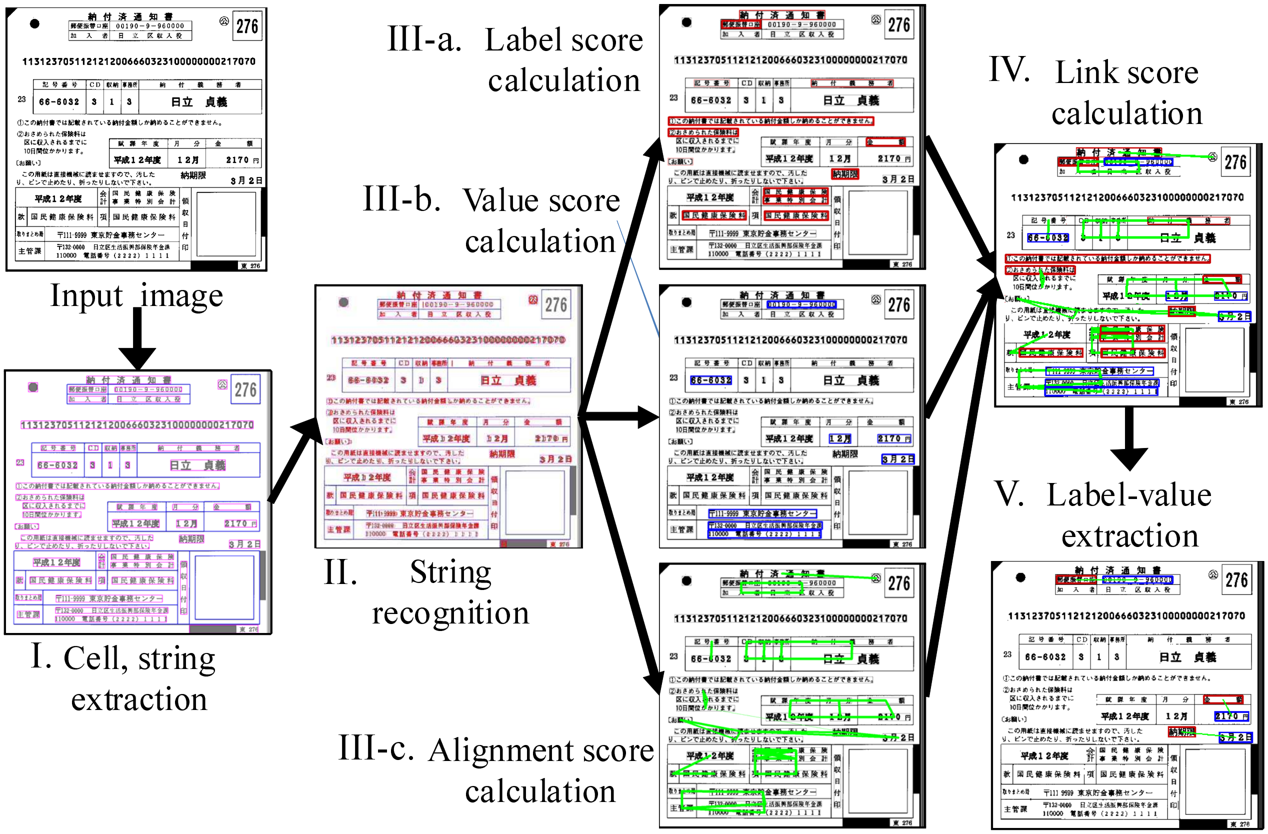

Input image
I. Cell, string extraction
II. String recognition
III-a. Label score calculation
III-b. Value score calculation
III-c. Alignment score calculation
IV. Link score calculation
V. Label-value extraction

Figure 5. Extraction flow of proposed method

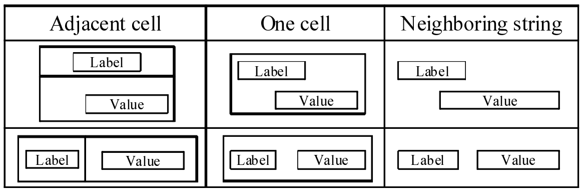


Figure 9. Alignment types

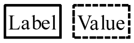
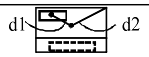
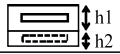
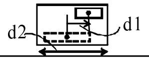
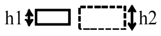
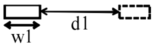
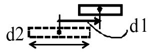
Alignment type	Penalty patterns of positional relation 		Penalty function
Adjacent cell	Label is located at edge of cell		$g1(i,j) = d1/d2$
	Height of frame: Label`s > Value`s		$g2(i,j) = h1/h2$ (If $h1 > h2$)
One cell	Label is located at right of value		$g3(i,j) = d1/d2$
Neighboring string	Height of strings are quite different		$g4(i,j) = h1/h2$ (If $h1 > h2$) $g4(i,j) = h2/h1$ (If $h1 < h2$)
	Label and value are distantly located		$g5(i,j) = d1/w1$ (If $d1 > 2 \times w1$)
	Label is located at right of value		$g6(i,j) = d1/d2$

Figure 10. Penalty patterns

ate for a label-value relation. To detect all the possible label-value relation candidates from an ambiguous layout, our method attempts to quantify the positional relation strength of every string pair (S_i, S_j).

Alignment patterns between two strings can be categorized into three types (see Fig. 9), i.e., adjacent cell relation, one cell relation, and neighboring string relation.

We defined penalty rules of positional relations. The penalty rules indicate the non-orderliness of two-string alignment as a label-value relation, using penalty function $g_k(i,j)$. Figure 10 shows examples of penalty rules.

We defined alignment score $Salign(S_i, S_j)$ by the following equation:

$$Salign(S_i, S_j) = 1 - \sum_{k=1}^{K} g_k(i,j) \quad (3)$$

where K is the number of penalty rules.

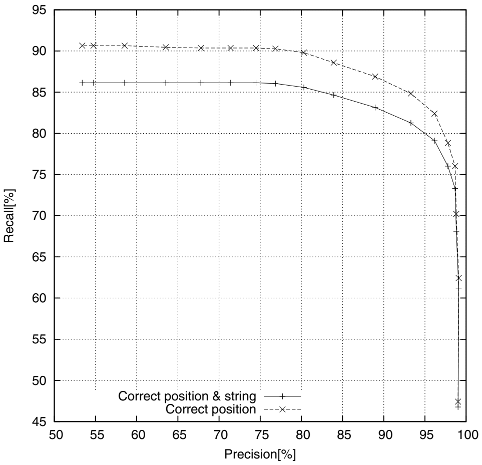


Figure 11. Recall-precision curve

D. Extraction of Label-value Relation

The label-value relation is determined by link score $Slink(S_i, S_j)$. If the $Slink(S_i, S_j)$ is greater than threshold TH_{link}, the string pair (S_i, S_j) is extracted as a label-value relation.

The link score is expressed by the following equation:

$$Slink(S_i, S_j) = \max_{\forall n,m} \Bigg\{ Salign(S_i, S_j) \times \big\{ Slabel(S_i, W_n) + Svalue(S_j, D_m) \big\} \Bigg| attr(W_n) = attr(D_m) \Bigg\}$$

where $attr(A)$ is a function that returns the attribute of dictionary A.

IV. Experiments

We evaluated our method on 200 form image samples of a payment slip (200 dpi, binary bitmap) that includes noisy characters and ambiguous alignment of the label-value relation. Out test samples are collected by scaning tax payment slips, which used in Japanese banks and local goverments. Extraction targets comprised three attributes: amount of payment, account number, and due date. The average of the number of strings on a form image is approximately 400 while those of target strings pairs is 5.5.

A recall-precision curve is shown in Fig. 11. The “Correct position” legend shows that target positions are successfully extracted and the “Correct position & string” legend shows that both positions and characters are successfully extracted. This graph shows that our method can achieve 85% recall rate and robustly extract label-value relations from noisy images and ambiguous alignment forms.

V. Conclusion

We have presented a “template-free form recognition” technique that extracts a label-value relation from various layout forms without pre-defined layout knowledge. Our method independently generates three different hypotheses (scores): i.e., label score, value score, and alignment score and finally determines a label-value relation by verifying all the possible combinations of these hypotheses. The evaluation demonstrated that our method maintains robustness for ambiguous layouts and noisy/low quality images.

References

[1] H. Shinjo, E. Hadano, K. Marukawa, Y. Shima, and H. Sako, *“A Recursive Analysis for Form Cell Recognition,”* Proc. of ICDAR2001, pp. 694-698, 2001.

[2] M. Seki, M. Fujio, T. Nagasaki, H. Shinjo, and K. Marukawa, *“Information Management System Using Structure Analysis of Paper/Electronic Documents and Its Applications,”* Proc. of ICDAR2007, pp. 689-693, 2007.

[3] A. Minagawa, Y. Fujii, H. Takebe, and K. Fujimoto, *“Logical Structure Analysis for Form Images with Arbitary Layout by Belief Propagation,”* Proc. of ICDAR2007, pp. 714-718, 2007.

[4] K. Taghva, R. Beckley, and J. Coombs, *“Extracting ”Carbon Copy” Names and Organizations From Heterogeneous Document Collection,”* Proc. of ICDAR2007, pp. 664-668, 2007.

[5] T. Watanabe, Q. Luo, and N. Sugie, *“Layout Recognition of Multi-Kinds of Table-Form Documents,”* IEEE Trans. on PAMI, vol.17, no.4, pp. 432-445, 1995.

[6] S. Marinai, *“Metadata Extraction from PDF Papers for Digital Library Ingest,”* Proc. of ICDAR2009, pp. 251-255, 2009.