Implementation and Evaluation of Similar Subgraph Retrieving

Hiroaki Kodama, Mitsuru Nakata, Qi-Wei Ge and Makoto Yoshimura
Faculty of Education, Yamaguchi University
1677-1 Yoshida Yamaguchi-shi, Yamaguchi Pref., Japan
Email: {w005km, mnakata, gqw, y_makoto}@yamaguchi-u.ac.jp

Abstract: We aim to realize an image retrieval system for handwritten Japanese historical documents, which doesn't require reprinting and revising processes. In this system, a feature graph is used to represent the structure of a string consisting of one or more characters. A feature graph of a string in historical document image is called document graph. And a feature graph of a string specified as a search condition by users is called search graph. In our system, document images including similar structure with a specified search graph are obtained by retrieving similar subgraphs of the search graph from a set of document graphs. Till now, we have realized a method of generating feature graphs and proposed an algorithm for similar subgraph retrieving. In this paper, we describe the implementation and evaluation of the similar subgraph search program.

Keywords—**similar subgraph retrieving, isomorphic subgraph, image retrieval for historical documents**

1. Introduction

In recent years, various Japanese historical documents stored in libraries and museums have been compiled into database and published on the Internet. To utilize these databases, effective search functions, i.e. keyword search and full text search, are required and historical documents image data must be stored in databases along with text data. Text data of documents are generated via reprinting and revising processes. Reprinting is to convert handwritten Japanese characters in historical documents into modern Japanese characters, and revising is to emend the reprinting results. These processes require a huge amount of time and high level of expertise. Therefore, there are many databases without text data. Websites with document images, linked from the document name and the page number, are the examples. Unfortunately these sites aren't utilized so much because no efficient search functions are provided. To solve this problem, some character recognition methods for handwritten character in historical documents have been proposed [1]-[3]. But there is still a problem that the accuracy of these methods isn't enough for practical use.

Under this background, we are to realize an image search system for Japanese historical document which needn't require reprinting and revising processes [4]. In our system, the structural information of a string written in historical documents is expressed by a feature graph. So far, we have realized a feature graph generating program and designed a similar subgraph retrieving algorithm for our document image search system [5]. In this paper, we describe the implementation and evaluation of the similar subgraph search algorithm.

2. Outline of our search system

Figure 1 shows the outline of our system. Structural information of a string is represented by "*feature graph*" [3]. Feature graph is a simple graph which expresses structure of a string in a document image. A vertex of the graph corresponds to a cross point, an end point or the most convex point of high curvature curve on a stroke. An edge represents a connection relationship between vertices. A feature graph includes coordinates of each vertex and the information of width and height of the image. The character structural information DB stores structural information of each line of historical documents together with information of the document image such as URL and size. The feature graph corresponding to each line of documents in the DB is called "*document graph*". Document graphs are converted from document images on the Internet by the input part. The condition generating part generates two feature graphs called "*search graph*" and "*essential graph*" as a search condition. Search graph is a feature graph which represents character structure for which users want to search. In other words, a search graph shows a search condition. Essential graph is a subgraph in a search graph. An essential graph consists of one or more connected components and represents the indispensable structure which must be included in every search results. The search part retrieves document graphs including similar structure to the search graph. The result output part displays historical document images corresponding to the retrieved document graphs.

3. Similar subgraph retrieving

3.1 Algorithm of the similar subgraph retrieving

The search part retrieves document graphs including subgraphs called "*similar subgraph*". The similar subgraph has a similar structure to a given search graph. Procedure $SimSubGraph$ is repeated for all document graphs by the search part. D=(V^D, E^D), S=(V^S, E^S) and I=(V^I, E^I) show a document graph, a search graph and an essential graph, respectively. These three graphs (D, S and I) are arguments of the procedure $SimSubGraph$. And the procedure obtains similar subgraphs in D. In the following, G_{CIR} expresses a similar subgraph having a similar structure with S. This procedure consists of the following three steps. Here, R_{CIR} expresses a rectangle showing a partial area corresponding to G_{CIR} in a historical document image. SIM is a similarity between S and G_{CIR}.

$\boldsymbol{StepI}$ **:** All isomorphic subgraphs of connected components of the essential graph I are obtained from a document graph D. Then a set consisting of combinations of these subgraphs is generated. Each element of the set is a result candidate

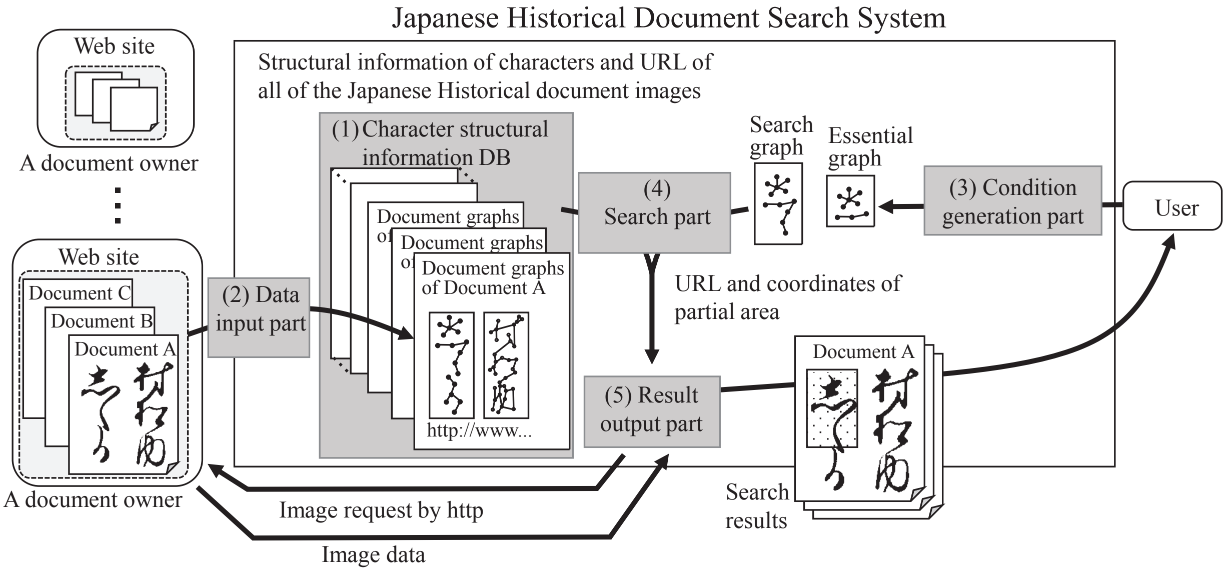

Figure 1. Outline of our Japanese historical document search system

of the procedure $SimSubGraph$. The essential graph I depicted in the left of Figure 2 consists of two connected components C_1 and C_2. The subgraph IC_1^1 in the document graph D is an isomorphic subgraph having the same structure to C_1. Likewise, IC_1^2–IC_{14}^2 are isomorphic subgraphs of C_2. Therefore, the result candidate set consists of (IC_1^1, IC_1^2), . . ., (IC_1^1, IC_{14}^2) as shown in Figure 2. Note that isomorphic subgraphs IC_2^2–IC_{11}^2, which overlap with IC_1^1, are omitted in the figure.

StepII **:** Inappropriate candidates are deleted from the result candidate set obtained in Step I. The criterion of the deletion is based on the number of vertices shared among subgraphs, position relation between subgraphs, relation among corresponding edges and so on. In the example shown in Figure 2, (IC_1^1, IC_2^2), . . ., (IC_1^1, IC_{11}^2) are deleted from the result candidate set because all vertices of IC_2^2–IC_{11}^2 are also vertices of IC_1^1. Although connected component C_1 of the essential graph is located above C_2, IC_1^1 corresponding to C_1 is located under IC_1^2 corresponding to C_2. Therefore, the candidate (IC_1^1, IC_1^2) is deleted. And also, (IC_1^1, IC_{13}^2) and (IC_1^1, IC_{14}^2) are deleted because the length and angle of corresponding edges among C_2 and IC_{13}^2 are quite different and IC_1^1 is too far from IC_{14}^2.

StepIII **:** For all remaining candidates, R_{CIR} is clipped from D based on layout and size of the connected components of each candidate. R_{CIR} expresses a rectangle expressing a partial area which has similar structure to S. The similarity SIM between S and G_{CIR} which consists of vertices and edges included in R_{CIR} is calculated. Finally, $RES{=}\{(R_{CIR}, G_{CIR}, SIM)\}$ is returned. G_{CIR} consists of vertices and edges of candidates obtained in Step II, and further, it might contain vertices and edges of D. These vertices and edges express "optional structure" which means inessential structure.

The similarity SIM is defiend by Equation (1).

$$SIM{=}(k_1{\cdot}SIM_A{+}k_2{\cdot}SIM_B){\cdot}SIM_C \tag{1}$$

Here, SIM_A expresses the similarity between an essential graph I and a subgraph IS. IS is isomorphic to I and it is a subgraph in D. SIM_B expresses the similarity between $(S{-}I)$ and $(G_{CIR}{-}IS)$. SIM_C is the size similarity between G_{CIR} and a document graph D. Also, $k_1{=}\frac{|v.I|+|v.IS|}{|v.S|+|v.G_{CIR}|}$ and $k_2{=}\frac{|v.S-v.I|+|v.G_{CIR}-v.IS|}{|v.S|+|v.G_{CIR}|}$. $v.I$, $v.IS$, $v.S$ and $v.G_{CIR}$ are vertex sets of I, IS, S and G_{CIR}, respectively. k_1 (k_2) expresses the proportion of vertices used in calculation of SIM_A (SIM_B) to vertices of G_{CIR} and S.

SIM_A is obtained according to Equation (2).

$$\begin{aligned} SIM_A &= 0.8{\cdot}S_1{+}0.2{\cdot}S_2 \\ S_1 &= a.INT/a.CIR \\ S_2 &= \frac{\sum_{i=1}^{n}\frac{\sum_{j=1}^{|Edg(IC_i)|}\gamma_{ij}}{|Edg(IC_i)|}}{n} \end{aligned} \tag{2}$$

Here, $0{<}S_1{\leq}1$, $0{<}S_2{\leq}1$. $a.INT$ and $a.CIR$ are the sizes of rectangle R_{INT} and R_{CIR}, respectively. R_{INT} is an intersectional part of R_i $(1{\leq}i{\leq}n)$. R_i is a rectangle area clipped from D according to $IC^i(1{\leq}i{\leq}n)$. n is the number of connected components of I. And R_{CIR} is a circumscribed rectangle of R_i $(1{\leq}i{\leq}n)$. S_2 is defined as the ratio of the corresponding edges to the edges of IS. γ_{ij} is defined by Equation (3).

$$\gamma_{ij} = \begin{cases} 1 & \text{if } Ang(e_j, f_j){\leq}30, \\ 0.5 & \text{else if } 30{<}Ang(e_j, f_j){\leq}60, \\ 0.2 & \text{otherwise.} \end{cases} \tag{3}$$

Here, e_j is an edge of C_i and f_j is an edge of IC^i. If e_j is a corresponding edge of f_j, $Ang(e_j,f_j)$ expresses an angle between edge e_j and f_j.

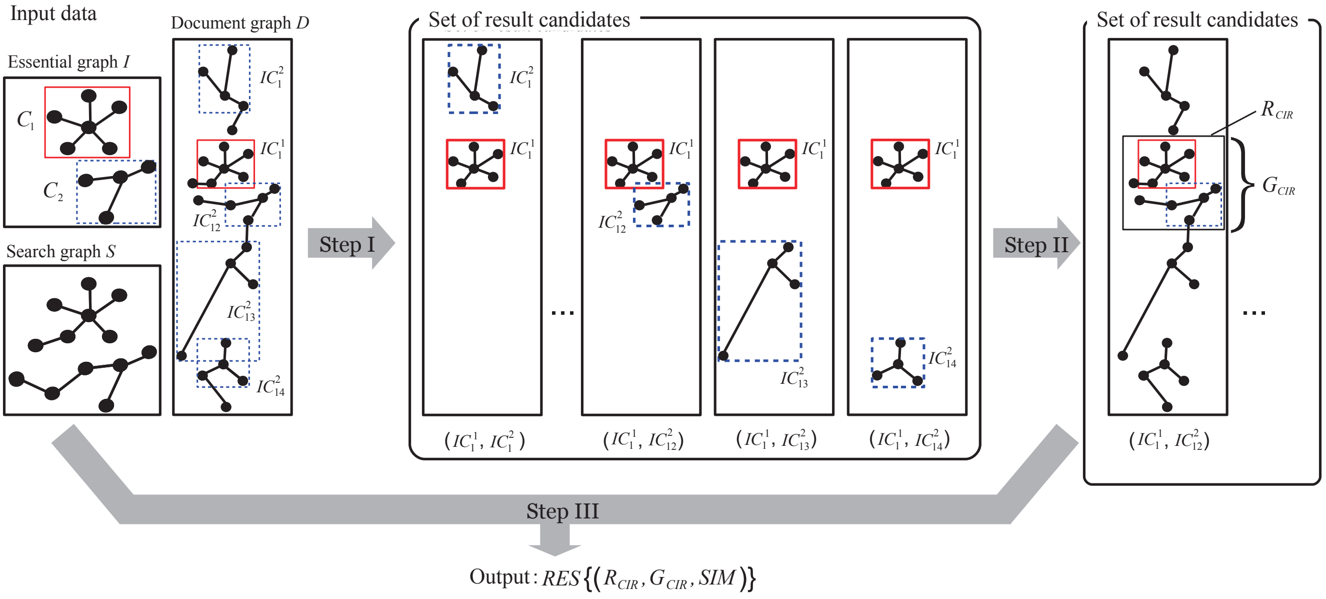

Figure 2. The process of the procedure $SimSubGraph$

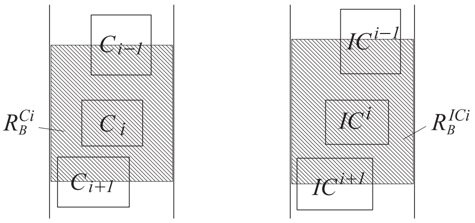

Figure 3. R_B^{Ci} and R_B^{ICi}

SIM_B is obtained according to Equation (4).

$$\begin{aligned} SIM_B &= (\sum_{i=1}^{n} SIM_B^i)/n & (4) \\ SIM_B^i &= (r_i^E + r_i^V)/2 \\ r_i^E &= func(|e.R_B^{Ci}|, |e.R_B^{ICi}|) \\ r_i^V &= (r_i^U + r_i^B + r_i^L + r_i^R)/4 \\ r_i^U &= func(|v^U.R_B^{Ci}|, |v^U.R_B^{ICi}|) \end{aligned}$$

Here, $func(a_1, a_2) = min\{a_1, a_2\} \times 2/(a_1 + a_2)$. R_B^{Ci} and R_B^{ICi} are rectangles based on C_i and IC^i, respectively. As shown in Figure 3, the top of R_B^{Ci} equals to the center of C_{i-1} and the bottom of R_B^{Ci} equals to the center of C_{i+1}. Furthermore, $e.R_B^{Ci}$ expresses a set of edges included in R_B^{Ci}, where $e.R_B^{Ci} \cap E^I = \phi$. So does $e.R_B^{ICi}$. $v^U.R_B^{Ci}$ expresses a set of virtices included in the upper half of R_B^{Ci}, where $v^U.R_B^{Ci} \cap V^I = \phi$. So do r_i^B, r_i^L, r_i^R. These correspond to the bottom, left and right half, respectively.

SIM_C $(0.9 < SIM_C \leq 1)$ is calculated according to Equation (5). Here, $W_{R_{CIR}}$ and W_D are the width of R_{CIR} and D, respectively.

$$SIM_C = 0.1 \cdot W_{R_{CIR}} / W_D + 0.9 \quad (5)$$

3.2 Implementation of the similar subgraph retrieving algorithm

We have made a Java program adopting the similar subgraph retrieving algorithm on a Windows PC (CPU: iCore7, Memory: 16Gbyte, OS: Windows 7 Professional). The java programming environment is Eclipse Ver.4.5 and JUNG (Java Universal Network/Graph Framework) Ver.1.7.6 is used as a graph-related library [6]. Under this computing environment, we have implemented the procedure $SimSubGraph$.

3.3 Evaluation experiment

We have conducted an experiment to evaluate our current program. Graphs S_A and S_B in Figure 4 are search graphs in the experiment. The dashed rectangles in the figure show essential graphs for each graph. And D_1, …, D_6 in Figure 5 are document graphs used in the experiment. These graphs are generated from string images clipped from *The Tale of Genji* or *Eiga Monogatari*. The execution results of the procedure $SimSubGraph$ are shown in Tables 1 and 2. As shown in Table 1, the total execution time for S_A and D_3 was 2.58 seconds. In this case, Step I, II and III took 1.50 seconds, 1.07 seconds and 0.01 seconds, respectively. The hyphen (-) in the table means that the corresponding step had not been executed. The 0.00 indicates that the execution time was less than 0.01 seconds. The average execution times for S_A and S_B were 19.1 seconds and 1.6 seconds. The result shows that the execution time of Step II is greatly affected by the structure of an essential graph. Table 2 shows the list of similarities between a search graph and a similar subgraph obtained by the procedure $SimSubGraph$. For example, G_{CIR}^1 is a similar subgraph retrieved from D_1. And the similarity between S_A and G_{CIR}^1 is 0.98.

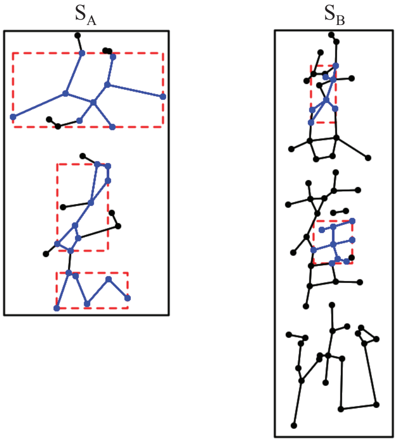

Figure 4. The given search graphs for the experiment

Table 1. The execution times of $SimSubGraph$

	S_A						S_B					
	D_1	D_2	D_3	D_4	D_5	D_6	D_1	D_2	D_3	D_4	D_5	D_6
Step I	0.67	1.07	1.50	1.38	3.42	1.44	0.44	0.52	0.71	0.90	1.63	0.87
Step II	0.78	-	1.07	-	103.3	-	0.08	-	0.02	1.18	0.88	1.97
Step III	0.00	-	0.01	-	-	-	-	-	-	0.10	0.03	0.21
Total	1.45	1.07	2.58	1.38	106.7	1.44	0.52	0.52	0.73	2.18	2.54	3.05

Table 2. The result of $SimSubGraph$

	G^1_{CIR}:S_A	G^3_{CIR}:S_A	G^4_{CIR}:S_B	G^5_{CIR}:S_B	G^6_{CIR}:S_B
SIM_A	0.99	0.35	0.97	0.75	0.58
SIM_B	1.00	0.62	0.95	0.93	0.83
SIM_C	0.98	1.00	1.00	1.00	1.00
SIM	0.98	0.47	0.95	0.89	0.78

3.4 Consideration

As shown in Figure 5, document graph D_2 has the similar structure as the search graph S_A (See the dashed circle in the figure). However, the procedure $SimSubGraph$ couldn't find it. The reason for this failure is that the shape of D_2 was incorrect. For example, many unnecessary vertices exist in the part indicated by the arrow in the circle. To solve this problem, the feature graph generation method should be improved.

In most cases, it took a few seconds to search for a similar subgraph from a document graph. To shorten this time, Steps I and II must be improved. In addition, when the structure of connected components of the essential graph is very simple, search time was extremely long and it was also unable to get the correct result. Therefore, an essential graph should have a sufficiently complex structure.

For retrieved similar subgraphs, the similarities were obtained almost correctly. In the future, we will conduct experiments with more document graphs and evaluate the validity of obtained similarities.

Acknowledgements: This work was partly supported by JSPS KAKENHI Grant Number 15K00469.

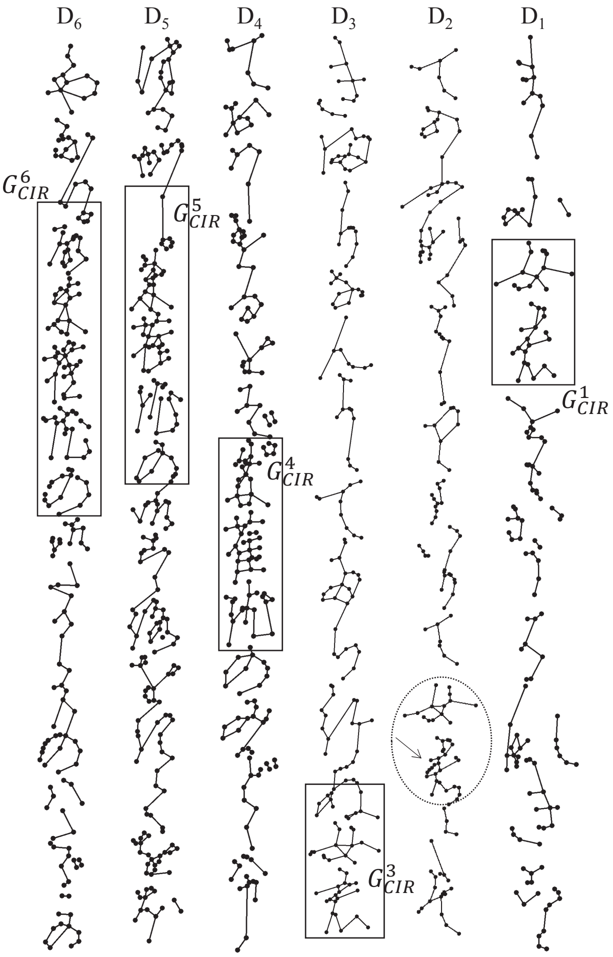

Figure 5. Six document graphs for the evaluation experiment

References

[1] Shoji Yamada, Mamoru Shibayama, "Outline of Historical Character Recognition Project", *information processing*, Vol.43, No.9, pp.950-955, 2002.

[2] Yuji Izumi, Nei Kato, Yoshiaki Nemoto, Shoji Yamada, Mamoru Shibayama, Hiroshi Kawaguchi, "A Study for Character Recognition of Ancient Documents using Neural Network",*IPSJ SIG Notes*, vol.2000-CH-45, pp.9-15, 2000.

[3] M.Nakata, S.Nishida, R.Fukuda, Qi-Wei Ge and M.Yoshimura, "A Method of Recognizing Handwritten Characters in Japanese Historical Documents by Using Feature Graphs", *INFORMATION*, vol.13, No.3(B), pp.953-966, 2010.

[4] Yuichiro Iino, Hiroaki Nagaoka, Mitsuru Nakata, Qi-Wei Ge, Makoto Yoshimura "A Feature Graph Based Retrieval System for Japanese Historical Documents", Proc. of *ITC-CSCC2014*, pp.982-985, 2014.

[5] Hiroaki Nagaoka, Yuichiro Iino, Mitsuru Nakata and Qi-Wei Ge "A Retrieval Method of Similar Subgraphs Used in Japanese Historical Documents Image Retrieval System", IEICE technical report 114(493), pp.71-76, 2015.

[6] *http://jung.sourceforge.net/*