Authors' Names Extraction from Scanned Documents

Manabu Ohta, Shun Yamasaki, Takayuki Yakushi
Okayama University
3-1-1 Tsushima-naka, Okayama-shi,
Okayama 700-8530, Japan
{ohta,yamasaki,yakushi }@de.it.okayama-u.ac.jp

Atsuhiro Takasu
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan
takasu@nii.ac.jp

Abstract

Authors' names are a critical bibliographic element when searching or browsing academic articles stored in digital libraries. However, extracting such bibliographic data from printed documents requires human intervention; it is therefore not cost-effective, even using various document image-processing techniques such as Optical Character Recognition (OCR). In this paper, we describe an automatic authors' names extraction method for academic articles scanned with OCR mark-up. The proposed method first extracts authors' blocks, which include assumed author/delimiter characters based on layout analysis, and then uses a specifically designed Hidden Markov Model (HMM) for labeling the unsegmented character strings in the block as those of either an author or a delimiter. We applied the proposed method to Japanese academic articles. Results of these experiments showed that the proposed method correctly extracted more than 99% of authors' blocks with manual tuning; the proposed HMM correctly labeled more than 95% of the author name strings.

1. Introduction

The digitizing of printed documents is an important task of constructing a large document archive. Document digitization has been studied intensely by the document image analysis community. Various tools and techniques have been developed, including noise reduction, deskewing, Optical Character Recognition (OCR) [1], page layout analysis [2], logical analysis, and so on. Automatic document image processing systems, which convert printed documents into their digital forms, have also been researched extensively as a result of improvements in document image analysis. For example, Taghva et al. [3,4] developed an automatic document mark-up system to mark words, sentences, paragraphs, and sections in OCR-scanned documents.

Digital libraries also require document analysis techniques [5,6] for accessing printed articles. They contain not only the full text but also metadata such as authors, titles, and references. As for references, several techniques for extracting reference fields from OCR outputs have been proposed because they are very useful once extracted [7,8]. CiteSeer [9], a famous autonomous citation indexing system, is a good example of using citations. In addition, authors' names are useful as keys for further retrieval and for recognizing the community to which they belong. Therefore, we have developed a title page analyzer to extract author names from title pages of articles. The document digitization process usually comprises several steps, such as OCR, layout analysis, and information extraction. Errors are inevitable in each step; error-tolerant document processing is highly demanded.

This paper proposes an authors' name extraction method that is both automatic and robust against OCR recognition errors. The input of our method is academic articles scanned using OCR mark-up (xml); its output is an augmented xml with author tags for each author. An important feature is that the proposed method simultaneously incorporates both recognition characteristics of the OCR used and syntactic constraints for strings in an authors' block. Thereby, some OCR errors are absorbed. Another feature is that the proposed method tunes its parameters using some amount of training data. Moreover, our method is useful because extracted authors' names are used for producing hypertext of scanned articles and the extracted names can be substituted for the metadata that are presently created through costly manual editing.

2. Authors' names extraction

The authors' names extraction problem is to extract lines representing authors in the title page of an academic article, and then to label every character in the lines as that of an *author* or *delimiter* so that each author can be extracted.

1-4244-1476-8/07/$25.00 ©2007 IEEE.

Fig. 1 shows an example of an authors' block of a title page, which usually starts with an author name followed by delimiters and continues with second or later authors. Delimiting characters, such as the dagger shown in Fig. 1, are used for specifying authors' affiliations in the footer. As Fig. 1 shows, we must extract four authors with removal of delimiters.

We have developed a title page analysis system in which authors' names are extracted from scanned images by the following steps:

1. page layout analysis and character recognition,
2. authors' block extraction, and
3. author/delimiter labeling.

2.1. Our OCR system

For page layout analysis and character recognition, we have developed an OCR system in collaboration with an OCR vendor. Japanese articles contain both English and Japanese words, which markedly degrades the recognition accuracy. For that reason, the OCR system includes both English and Japanese OCR engines. It first determines the dominant language of each line in an article, then selects an engine according to the dominant language. For each scanned page, the system produces recognized text with the bounding rectangles for characters, lines, and blocks.

2.2. Authors' block extraction

An authors' block, a kind of logical block, includes at least a physical block produced by the OCR system and might constitute several blocks. The authors' area typically follows the title area and is followed by the abstract area, although different journals have slightly different layouts. In addition to these, we use several heuristic rules to determine authors' blocks. They are based on the vertical position of lines as follows:

1. Authors' lines appear in a certain vertical range.
2. The title and authors are typically separated vertically by a gap, as are authors and the abstract.

To determine these thresholds, we use a small amount of data as a gold standard of manually tagged data with explicit title, authors', and abstract block boundaries. Thus, we extract authors' blocks automatically using these thresholds learned from the data. However, because the character string might contain OCR recognition errors, we propose a Hidden Markov Model (HMM), which can handle them statistically. We describe this model next.

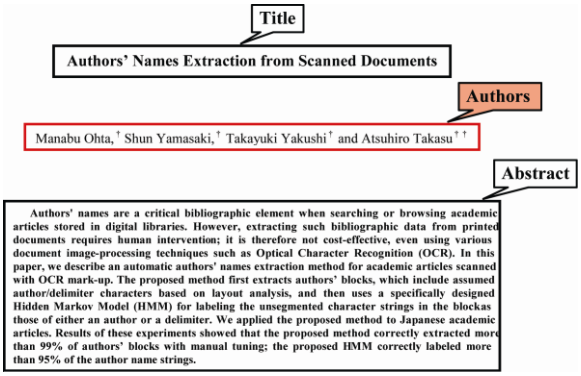

Figure 1. Authors' block example in a title page

2.3. Author/delimiter labeling

The characters in the authors' block comprise name and delimiting characters. Of course, they contain OCR misrecognitions. For that reason, we propose a specifically designed HMM for labeling these characters as authors or delimiters. This model gives the most likely labeling when given a character string considering both length and characters of name or delimiter. In this model, we assume that name and delimiter strings appear alternately and that expectations of character appearances are determined depending on the character position in each string.

The proposed model is a form of HMM. It produces a string by traversing the finite states which are characterized by a label indicating either author or delimiter in addition to the character position in each string.

Fig. 2 shows an example of the proposed HMM. States *s* and *e* respectively represent *start* and *end*. Therefore, $\pi(s)=1$ and *e* is the last state where $\pi(s)$ denotes the initial probability. A state *ai* represents *i*th position of an author name string, and a state *dj* does *j*th position of a series of delimiters. Variables *x* and *y* denote the maximum length of name and delimiter strings, respectively, and can be learned from training data.

This HMM has transition probabilities, which are denoted by arcs in Fig. 2. Based on our assumptions, the state transitions have the following restrictions.

1. One can move from *ai* only to *a(i+1)*, *d1*, or *e*.
2. One can move from *dj* only to *d(j+1)*, *a1*, or *e*.

In those expressions of restrictions, $1 \le i \le x-1$ and $1 \le j \le y-1$. In addition, as shown in Fig. 2, one can move from *ax* to *ax* and *dy* to *dy* to handle long names and delimiting strings, neither of which appear in the training data. This achieves a kind of smoothing for transition probabilities.

Each state produces a symbol (character) according to an output probability distribution. Presuming that the alphabet is $\{u, v\}$, the states *ai*

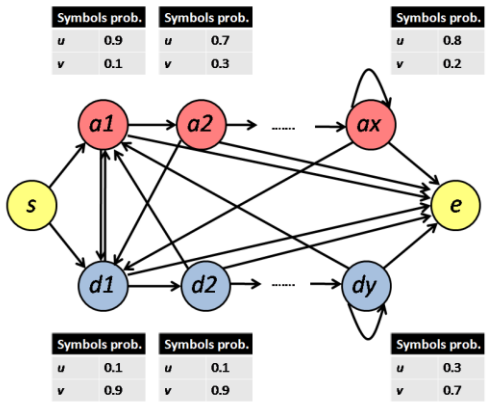

Figure 2. Proposed HMM

and *dj* produce a character *u* or *v* according to the output probabilities, which are shown in tables attached to states in Fig. 2. Both states *s* and *e* output no characters. In this model, the OCR errors are handled probabilistically, which means that delimiting characters can be output at states *ai* and normal characters output at states *dj* according to the error distribution of training data. That is, if the training data include some misrecognitions, then the output probability reflects them. In this way, the OCR errors are absorbed in this labeling step.

For a character string α, which is a mixture of author names and delimiters, assume that the proposed HMM produces α by a state transition $\mathbf{q} \equiv q_1 q_2 \cdots q_t$, and that it produces a character α_i at each state $q_i \in \mathbf{q}$ $(1 \le i \le t-1)$, where $q_1 = s$, $q_t = e$, $\alpha_1 = \phi$, and ϕ stands for the null character. Then, the HMM M gives the probability of outputting α with the state transition $\mathbf{q}$ as

$$P(\alpha, \mathbf{q} \mid M) = \prod_{i=1}^{t-1} [o(\alpha_i, q_i) \times \tau(q_i, q_{i+1})],$$

where $o(\alpha_i, q_i)$ and $\tau(q_i, q_{i+1})$ respectively denote the output and transition probabilities.

For a string α, let $\mathbf{Q}(\alpha)$ denote the set of state transitions of the HMM that produce α. Then, the state transition giving the highest probability that the HMM will produce α is

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q} \in \mathbf{Q}(\alpha)} P(\alpha, \mathbf{q} \mid M).$$

We can obtain such a state transition using the Viterbi algorithm. This state transition is the very sequence of author/delimiter labels for a given string α because it comprises either *ai* or *dj*, except for the first and last states of *s* and *e*.

Suppose a character string α = "1stAuthor†,*2ndAuthor‡" is obtained as a result of the authors' block extraction. String α is correctly labeled if the Viterbi algorithm outputs a state transition *s-a1-a2-a3-a4-a5-a6-a7-a8-a9-d1-d2-d3-a1-a2-a3-a4-a5-a6-a7-a8-a9-d1-e* as the most likely sequence of hidden states. Here, the state subsequence *d1-d2-d3* corresponds to the intermediate delimiting characters "†,*", and the next-to-last state *d1* to the last delimiter "‡". The (sub)string "1stAuthor" is output during the transition of the former state subsequence *a1-a2-a3-a4-a5-a6-a7-a8-a9*, whereas the string "2ndAuthor" is output during the transition of the latter. Even if a string within the authors' block includes OCR errors, we can expect correct labeling because the trained HMM can be expected to take these errors into consideration. For ease of explanation, we provided an English example here, but our method mainly targets Japanese academic articles now.

3. Experiments

We apply the proposed method to OCR-processed academic articles that have appeared in journals. The task is divisible into two parts: i) to extract authors' blocks from the articles, and ii) to label the strings within the block as those of an author or delimiter. For the experiment, we used 54 issues published in two years, 2003 (vol. 44) and 2004 (vol. 45), by the Information Processing Society of Japan (IPSJ), a famous academic society of computer science in Japan.

3.1. OCR recognition accuracy

We first scan papers at 400 pixel/inch (ppi) resolution and then apply the developed OCR system to extract the text. We choose papers randomly from the data set and measure the accuracy of the OCR for the abstract and references to check the OCR performance. The recognition accuracy of the abstract is 99.00%, but that of the references is 97.01%.

The accuracy for the references is lower than that for the abstracts. Three main reasons pertain.

1. The reference strings typically contain both Japanese and English characters, whereas the abstracts contain mainly Japanese.
2. The reference strings contain various fonts, and the OCR tends to recognize italic fonts incorrectly.
3. The reference strings contain various punctuation symbols, and the OCR tends to confuse symbols such as colons, semicolons, commas and periods.

3.2. Performance of authors' block extraction

From the OCR-processed mark-up text, we first extract authors' blocks. In other words, one or more

blocks generated by the OCR are labeled as authors if they contain the name strings of all authors.

In the experiment, we used 290 articles in vol. 44 as a training set to determine the thresholds described in Section 2.2. Table 1 shows the experimental results. We correctly extracted 205 of 243 authors' blocks from the vol. 45 test set. Table 1 also shows that the authors' blocks of 241 articles were extracted correctly using thresholds determined by humans referring to 173 articles collected from other volumes. One extraction error was caused by noise located between the authors and abstract; consequently, the authors and abstract were not properly separable. In the other case, the second line of a title was extracted as authors by mistake.

The experiment shows that the proposed heuristic rules showed greater than 99% extraction accuracy with properly determined thresholds. Therefore, determination of such thresholds with small samples is to be solved.

3.3. Performance of author/delimiter labeling

For HMM construction and performance evaluation, we need a gold standard of correctly labeled OCR-markup with explicit indication of each author. We developed a program to produce the standard using manually created metadata derived from outputs of a bibliographic database. As training and test data, we chose articles having correctly extracted authors' blocks and the following properties. That is, each author string included in the article has at most one substitution error and at most *n-1* insertion errors, where *n* is the name string length. Table 2 summarizes the data statistics. We used 361 articles of vol. 44 as training data and then used 323 of vol. 45 for evaluating the performance of the proposed HMM.

To label the characters in authors' blocks, we first construct a proposed HMM using the training data. We obtained 9 author and 11 delimiter states, i.e., *x* and *y* described in Section 2.3 were, respectively, 9 and 11. State *a1* was estimated to be able to output 265 different characters, whereas state *d1* output only 25 as a training result. The other experimental conditions were as follows.

Table 1. Extraction accuracy of authors' blocks

Threshold	# Extracted	Accuracy(%)
Training	205	84.36
Manual	241	99.18

1. At states *ai* and *dj,* all output probabilities whose values were estimated as 0 by training were assigned 0.0000001 ('flooring').
2. As for transition probabilities, we interpolated only states *a9* and *d11*, which were the last states of the author and delimiter, respectively. That is, we assigned 0.2 for self-transitions, 0.1 for transitions *a9* to *d1* and *d11* to *a1*; the rest (0.7) were divided according to training.

Tables 3 and 4 respectively summarize the resultant labeling accuracy using the proposed HMM when extracting each author string from training and test data. Table 4 shows that the proposed method correctly labeled 1042 of 1092 authors: the labeling accuracy was 95.42%. With respect to the number of articles, 273 articles were correctly labeled; its accuracy was 84.52%.

Labeling errors are typically found between name and delimiter strings and often involve OCR errors. For example, all the errors in Table 3 resulted from OCR misrecognitions of characters or noise of scanned documents located at the end of name strings, i.e., at the beginning of succeeding delimiter strings. Moreover, the first or last character of name strings in test data was sometimes mistakenly labeled as a delimiter, even if they were recognized correctly. This is considered to be attributable to the characters' non-appearance in training data.

Through the experiments, we also found five errors with manually created metadata, which is usually considered as error-free. Five missing name strings were found in articles that had many co-authors. The proposed method can detect such an error with metadata by finding consecutive labeling errors greater than a certain length because the whole string of one author is mistakenly labeled as a delimiter in the presumed gold standard, but it is correctly labeled by the HMM. In contrast, other labeling errors were of, at most, two consecutive characters and of a single character in most cases.

Unused articles shown in Table 2 are those including other types of errors such as deletions, and compounded and ambiguous errors. Those including ambiguous errors tend to have bad quality because of poor scanning and appear intensively in a particular issue of a journal. We should rescan and re-recognize such articles rather than handle them by the proposed method as they are. By contrast, the other articles

Table 2. Number of articles included in the data

Vol.	Total	Used	Unused
44	511	361	150
45	425	323	102

Table 3. Labeling accuracy for training data

	Total	Extracted	Ratio(%)
# Article	361	342	94.74
# Author	1172	1153	98.38

Table 4. Labeling accuracy for test data

	Total	Extracted	Ratio(%)
# Article	323	273	84.52
# Author	1092	1042	95.42

should be processed using the proposed method. However, we should mind the tradeoff between applicability and accuracy when using such articles as training data.

We can find few works related to bibliographic data extraction from a title page of scanned academic articles, but some from other parts of scanned articles are available. Takasu et al. proposed a robust reference extraction method from scanned documents and applied it to articles appearing in various journals [7]. They first extracted reference sections and then references, which is procedurally similar to our block extraction and labeling steps. Their method is also based on a kind of HMM, which can handle substitution, deletion, and insertion errors. They experimented on the same journal but on different issues from ours and achieved 89.99% of reference extraction accuracy when OCR recognition accuracy was 97.58%. This extraction accuracy is the product of both reference section and reference extractions. Although reference extraction includes some intrinsically different problems described in Section 3.1, our results shown in Tables 1 and 4 are still considered to be competitive and practical.

4. Conclusions and future works

This paper proposes an automatic extraction method for authors' names from recognized documents. The proposed method first extracts an authors' block from each title page and then applies the special HMM to label the characters within the block as those of authors or delimiters. The proposed HMM defines the average length of author/delimiter strings by transition probabilities. It also defines the probabilities of outputting characters, given the position of either string of the author or delimiter, using output probabilities. It comprises few states: 22 including *s* and *e* states in our experiments. Consequently, it is easy to handle. The experiments for authors' block extraction show that simple heuristic rules achieved an extraction accuracy of 99.18% with manually tuned parameters and 84.36% with those learned from samples. Moreover, the experiments for labeling, i.e., each author extraction, shows that more than 95% of author name strings were extracted from test data. Consequently, the proposed HMM can handle OCR recognition errors; this ability reduces extraction errors.

To improve the extraction accuracy, we plan to use other characteristics of recognized characters, such as the width or height of their bounding rectangles. As Fig. 1 shows, such information is especially useful for Japanese articles because delimiting characters are typically superscripts and smaller than normal characters used for name strings. Therefore, we would like to integrate information of characters and bounding rectangles and make them available for the proposed HMM in the future.

We have also been developing a title page analysis system in which bibliographic elements, not only authors, but title, abstract, affiliations, etc., are extracted from scanned documents semi-automatically and error-free metadata can be generated with as little human intervention as possible. We need an error detector that has low miss and false alarm rates to reduce the cost of subsequent manual correction. We plan to develop such an error detector and a method for estimating these rates to assure the quality of the analysis system.

References

[1] H. Bunke and P. Wang, editors. *Handbook of Character Recognition and Document Image Analysis*. World Scientific, 1997.

[2] G. Nagy, S. Seth, and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals", *IEEE Computer*, Vol.25, No.7, pp.10-22, 1992.

[3] K. Taghva, A. Condit, and J. Borsack, "An Evaluation of an automatic markup system", In *Proc of IS&T/SPIE 1995 Intl. Symp. on Electronic Imaging Science and Technology*, pp.317-327, 1995.

[4] K. Taghva, A. Condit, J. Borsack, J. Kilburg, C. Wu, and J. Gilbreth, "The MANICUE document processing system", *Technical Report 95-02, Information Science Research Institute, University of Nevada*, 1995.

[5] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document Analysis System", *IBM Journal of Research and Development*, Vol.26, No.6, pp.647-656, 1982.

[6] G. A. Story, L. O'Gorman, D. Fox, L. L. Schaper, and H. V. Jagadish, "The RightPages Image-based Electronic Library for Alerting and Browsing", *IEEE Computer*, Vol.25, No.9, pp.17-26, 1992.

[7] A. Takasu and K. Aihara, "Quality Enhancement in Information Extraction from Scanned Documents", In *Proc. of DocEng'06*, pp.122-124, 2006.

[8] F. Parmentier and A. Belaid, "Bibliography References Validation Using Emergent Architecture", In *Proc. of IAPR International Conference on Document Analysis and Recognition*, pp.532-535, 1995.

[9] C. L. Giles, K. D. Bollacker, and S. Lawrence, "CiteSeer: An Automatic Citation Indexing System", In *Proc. of International Conference on Digital Libraries*, pp.89-98, 1998.