Detection and Analysis of Table of Contents Based on Content Association

Xiaofan Lin, Yan Xiong[1]
Imaging Systems Laboratory
HP Laboratories Palo Alto

HPL-2005-105
May 25, 2005*

table of contents, document structure analysis, table recognition, optical character recognition, algorithm combination

As a special type of table understanding, the detection and analysis of tables of contents (TOCs) play an important role in the digitization of multi-page documents. Most previous TOC analysis methods only concentrate on the TOC itself without taking into account the other pages in the same document. Besides, they often require manual coding or at least machine learning of document-specific models. This paper introduces a new method to detect and analyze TOCs based on content association. It fully leverages the text information throughout the whole multi-page document and can be directly applied to a wide range of documents without the need to build or learn the models for individual documents. In addition, the associations of general text and page numbers are combined to make the TOC analysis more accurate. Natural language processing and layout analysis are integrated to improve the TOC functional tagging. The applications of the proposed method in a large-scale digital library project are also discussed.

* Internal Accession Date Only

[1] KLA-Tencor Corporation, One Technology Drive, MS 1-2242, Milpitas, CA 95035 USA

To be published in the International Journal on Document Analysis and Recognition, 2005
Approved for External Publication

Detection and Analysis of Table of Contents Based on Content Association

Xiaofan Lin[1] and Yan Xiong[2]
[1] Hewlett-Packard Laboratories
1501 Page Mill Road, MS 1203, Palo Alto, CA 94087, USA
[2] KLA-Tencor Corporation
One Technology Drive, MS 1-2242, Milpitas, CA 95035, USA

Abstract

As a special type of table understanding, the detection and analysis of tables of contents (TOCs) play an important role in the digitization of multi-page documents. Most previous TOC analysis methods only concentrate on the TOC itself without taking into account the other pages in the same document. Besides, they often require manual coding or at least machine learning of document-specific models. This paper introduces a new method to detect and analyze TOCs based on content association. It fully leverages the text information throughout the whole multi-page document and can be directly applied to a wide range of documents without the need to build or learn the models for individual documents. In addition, the associations of general text and page numbers are combined to make the TOC analysis more accurate. Natural language processing and layout analysis are integrated to improve the TOC functional tagging. The applications of the proposed method in a large-scale digital library project are also discussed.

Key words: table of contents, document structure analysis, table recognition, optical character recognition, algorithm combination

1 Introduction

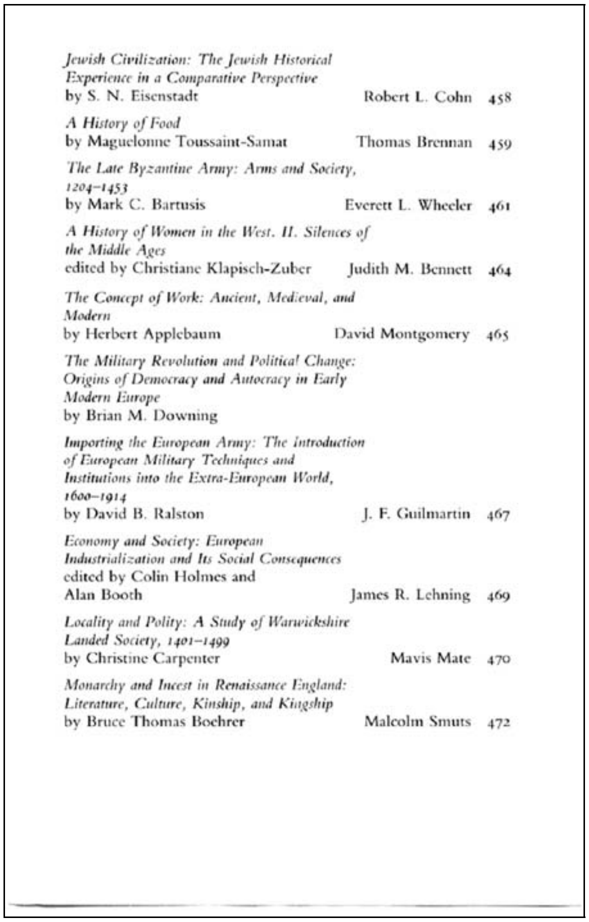
There is a high demand for the automatic conversion of printed multi-page documents into electronic format. For example, the initial steps in most digital library projects involve the digitization and recognition of large collections of multi-page documents such as journals, magazines, and books. Besides the recognition of individual pages using OCR software, it is usually desirable to organize the logical units (for example, articles in journals and chapters in books) of a document into a systematic structure to facilitate future information retrieval. Fortunately, most multi-page documents come with a built-in table of contents (TOC), which naturally reflects the logical structure of the entire document. The TOC describes each constituent article in a concise format: the title, author(s), and start page number. Thus, an efficient approach to multi-page document structure analysis is detecting and analyzing TOC pages.

A number of methods on TOC detection/analysis have been reported in the literature. As the following summary indicates, those methods usually require the creation of specific models for the documents to be processed. The simplest approach is to hard-code this *a priori* knowledge as rules into a system. For example, layout patterns or special text

* The second author conducted the research related to this paper while working for Hewlett-Packard Laboratories.

sequences can be used to segment article reference blocks and to extract fields related to each article. In order to analyze the logical structure of books in Japanese, Lin et al. [2] introduced a system of TOC page analysis and logical structure extraction by layout modeling and headline matching. They listed four patterns of text lines on the TOC pages. He et al. [10] combined geometrical rules (indentations) and semantic rules (typical text sequences identifying chapters and sections) to extract hierarchical logical structure in Chinese books. Mandal and Chowdhury [6] used layout rules to detect and segment the TOC pages from the University of Washington document database. Tsuruoka et al. [7] constructed models based on indentation features to extract structural elements such as chapters and sections in a book. The RightPages system developed by Story et al. [5] depended on layout models handcrafted for individual journal titles. Instead of building large number of layout models needed for various TOC pages, Belaïd [1] formulated several POS tagging rules to delimit articles. Realizing the heavy burden of hard-coding the rules, some researchers adopted machine learning to automate the model generation process. In the CyberMagazine project, Satoh et al. [3] proposed a system where TOC pages of academic journals were converted into bibliographic database by image segmentation. They used training data to learn decision trees for different kinds of journals. Bourgeois et al. [8] proposed a probabilistic relaxation method to analyze the structure of periodicals through training on the layouts of several representative samples.

The prerequisite of a layout modeling process will seriously limit the practical value of the system. As shown in Figure 1, there are so many possible layouts for TOC pages that it is almost impossible to exhaustively model all of them. Thus, it is quite likely that the TOC understanding system created for one document collection cannot be used on another collection. Learning from examples only provides a partial solution for certain applications such as journal processing, in which a large number of documents in the same style are expected. Even so, it still requires the careful selection of good samples and manual ground truthing, and thus can become a bottleneck in the whole workflow. Using POS tagging alone also carries limitations. It is language dependent and thus different POS models are required for different languages. In addition, the presence of OCR errors can distort the targeted POS patterns.



Jewish Civilization: The Jewish Historical Experience in a Comparative Perspective
by S. N. Eisenstadt Robert L. Cohn 458

A History of Food
by Maguelonne Toussaint-Samat Thomas Brennan 459

The Late Byzantine Army: Arms and Society, 1204–1453
by Mark C. Bartusis Everett L. Wheeler 461

A History of Women in the West. II. Silences of the Middle Ages
edited by Christiane Klapisch-Zuber Judith M. Bennett 464

The Concept of Work: Ancient, Medieval, and Modern
by Herbert Applebaum David Montgomery 465

The Military Revolution and Political Change: Origins of Democracy and Autocracy in Early Modern Europe
by Brian M. Downing

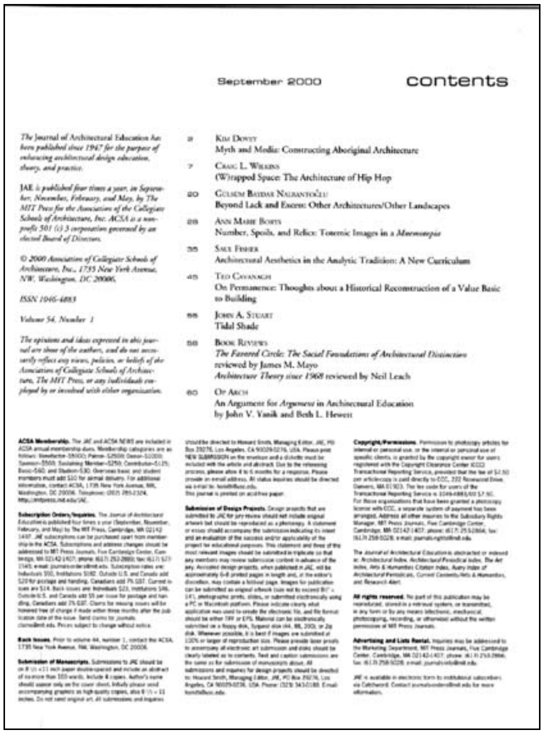
Importing the European Army: The Introduction of European Military Techniques and Institutions into the Extra-European World, 1600–1914
by David B. Ralston J. F. Guilmartin 467

Economy and Society: European Industrialization and Its Social Consequences
edited by Colin Holmes and Alan Booth James R. Lehning 469

Locality and Polity: A Study of Warwickshire Landed Society, 1401–1499
by Christine Carpenter Mavis Mate 470

Monarchy and Incest in Renaissance England: Literature, Culture, Kinship, and Kingship
by Bruce Thomas Boehrer Malcolm Smuts 472

Figure 1(a)



September 2000 contents

The Journal of Architectural Education has been published since 1947 for the purpose of enhancing architectural design education, theory, and practice.

JAE is published four times a year, in September, November, February, and May, by The MIT Press for the Association of Collegiate Schools of Architecture, Inc. ACSA is a nonprofit 501 (c) 3 corporation governed by an elected Board of Directors.

© 2000 Association of Collegiate Schools of Architecture, Inc., 1735 New York Avenue, NW, Washington, DC 20006.

ISSN 1046-4883

Volume 54, Number 1

The opinions and ideas expressed in this journal are those of the authors, and do not necessarily reflect any views, policies, or beliefs of the Association of Collegiate Schools of Architecture, The MIT Press, or any individuals employed by or involved with either organization.

3 Kim Dovey
Myth and Media: Constructing Aboriginal Architecture

7 Craig L. Wilkins
(W)rapped Space: The Architecture of Hip Hop

20 Gülsüm Baydar Nalbantoğlu
Beyond Lack and Excess: Other Architectures/Other Landscapes

28 Ann Marie Borys
Number, Spoils, and Relics: Totemic Images in a *Mnemotopia*

35 Saul Fisher
Architectural Aesthetics in the Analytic Tradition: A New Curriculum

45 Ted Cavanagh
On Permanence: Thoughts about a Historical Reconstruction of a Value Basic to Building

55 John A. Stuart
Tidal Shade

58 Book Reviews
The Favored Circle: The Social Foundations of Architectural Distinction reviewed by James M. Mayo
Architecture Theory since 1968 reviewed by Neil Leach

60 Op Arch
An Argument for *Argument* in Architectural Education
by John V. Vanik and Beth L. Hewett

ACSA Membership. The JAE and ACSA NEWS are included in ACSA annual membership dues. Membership categories are as follows: Benefactor–$5000; Patron–$2500; Donor–$1000; Sponsor–$500; Sustaining Member–$250; Contributor–$125; Basic–$60; and Student–$30. Overseas basic and student members must add $10 for airmail delivery. For additional information, contact ACSA, 1735 New York Avenue, NW, Washington, DC 20006. Telephone: (202) 785-2324. http://mitpress.mit.edu/JAE.

Subscription Orders/Inquiries. The Journal of Architectural Education is published four times a year (September, November, February, and May) by The MIT Press, Cambridge, MA 02142-1407. JAE subscriptions can be purchased apart from membership in the ACSA. Subscriptions and address changes should be addressed to MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1407; phone: (617) 253-2889; fax: (617) 577-1545; e-mail: journals-orders@mit.edu. Subscription rates are: Individuals $50, Institutions $182. Outside U.S. and Canada add $20 for postage and handling. Canadians add 7% GST. Current issues are $14. Back issues are: Individuals $23, Institutions $46. Outside U.S. and Canada add $5 per issue for postage and handling. Canadians add 7% GST. Claims for missing issues will be honored free of charge if made within three months after the publication date of the issue. Send claims to journals-claims@mit.edu. Prices subject to change without notice.

Back Issues. Prior to volume 44, number 1, contact the ACSA, 1735 New York Avenue, NW, Washington, DC 20006.

Submission of Manuscripts. Submissions to JAE should be on 8 1/2 x 11 inch paper double-spaced and include an abstract of no more than 100 words. Include 4 copies. Author's name should appear only on the cover sheet. Initially please send accompanying graphics as high-quality copies, also 8 1/2 x 11 inches. Do not send original art. All submissions and inquiries should be directed to Howard Smith, Managing Editor, JAE, PO Box 29276, Los Angeles, CA 90029-0276, USA. Please print NEW SUBMISSION on the envelope and a diskette must be included with the article and abstract. Due to the refereeing process, please allow 4 to 6 months for a response. Please provide an email address. All status inquiries should be directed via e-mail to: [illegible].
This journal is printed on acid-free paper.

Submission of Design Projects. Design projects that are submitted to JAE for jury review should not include original artwork but should be reproduced as a photocopy. A statement or essay should accompany the submission indicating its intent and an evaluation of the success and/or applicability of the project for educational purposes. This statement and three of the most relevant images should be submitted in triplicate so that any members may review submission content in advance of the jury. Accepted design projects, when published in JAE, will be approximately 6-8 printed pages in length and, at the editor's discretion, may contain a foldout page. Images for publication can be submitted as original artwork (size not to exceed 8 1/2" x 14"), photographic prints, slides, or submitted electronically using a PC or Macintosh platform. Please indicate clearly what application was used to create the electronic file, and file format should be either TIFF or EPS. Material can be electronically submitted on a floppy disk, Syquest disk (44, 88, 200), or Zip disk. Whenever possible, it is best if images are submitted at 100% or larger of reproduction size. Please provide laser proofs to accompany all electronic art submission and disks should be clearly labeled as to contents. Text and caption submissions are the same as for submission of manuscripts above. All submissions and inquiries for design projects should be directed to: Howard Smith, Managing Editor, JAE, PO Box 29276, Los Angeles, CA 90029-0276, USA. Phone: (323) 343-0188. E-mail: [illegible].

Copyright/Permissions. Permission to photocopy articles for internal or personal use, or the internal or personal use of specific clients, is granted by the copyright owner for users registered with the Copyright Clearance Center (CCC) Transactional Reporting Service, provided that the fee of $2.50 per article-copy is paid directly to CCC, 222 Rosewood Drive, Danvers, MA 01923. The fee code for users of the Transactional Reporting Service is 1046-4883/00 $2.50. For those organizations that have been granted a photocopy license with CCC, a separate system of payment has been arranged. Address all other inquiries to the Subsidiary Rights Manager, MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1407; phone: (617) 253-2864; fax: (617) 258-5028; e-mail: journals-rights@mit.edu.

The Journal of Architectural Education is abstracted or indexed in: Architectural Index, Architectural Periodical Index, The Art Index, Arts & Humanities Citation Index, Avery Index of Architectural Periodicals, Current Contents/Arts & Humanities, and Research Alert.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the written permission of MIT Press Journals.

Advertising and Lists Rental. Inquiries may be addressed to the Marketing Department, MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1407; phone: (617) 253-2866; fax: (617) 258-5028; e-mail: journals-info@mit.edu.

JAE is available in electronic form to institutional subscribers via Catchword. Contact journals-orders@mit.edu for more information.

Figure 1(b)

Figure 1: Different TOC layout styles

This paper presents a new method that can detect and analyze the TOC pages in a wide range of multi-page documents without explicit modeling or training. For the TOC detection and article linking stage (in this paper "article" means both an article in a journal/magazine and a chapter in a book), we only make one very universal assumption: Some contents (title, author name(s), and start page number, etc.) of an article will repeat on the title page (the first page of an article). In fact, this method is rooted in the fundamental nature of TOC: A TOC is simply a collection of references to individual articles of a document no matter what layout it follows. By taking full advantage of this characteristic, it is possible to find a widely applicable TOC detection/analysis solution. Along this direction, an efficient algorithm is designed to associate both the general text contents and the page numbers on TOC pages with those on the body (non-TOC) pages. Such association simultaneously solves the TOC detection and article linking problems. Besides, layout information and POS information are fused in the TOC functional tagging stage to make the system more robust.

Section 2 describes the overall processing workflow. Section 3 discusses the core TOC detection and article linking algorithms. Section 4 introduces the article separation algorithm. Section 5 presents the TOC functional tagging algorithm. Sections 4 and 5 also cover the experimental results. Section 6 contains a summary and points to the directions of future work.

2 Overall workflow

The document processing system is developed for the Digital Content Re-Mastering (DCRM) research project under collaboration between HP Labs and MIT Press [22]. The goal is to convert MIT Press' three thousand out-of-print books, journals, and magazines from paper to high-quality electronic version to enables new services such as on-line reading and print-on-demand (POD). Figure 2 depicts the overall workflow of our system as well as the interaction between the TOC Detection and Analysis System (TOCDAS) and other components. The first step is to scan all pages of a document into color/gray-scale TIFF files. Commercial OCR software development kit (SDK) is used to recognize the images and to generate the text and word bounding boxes. Meanwhile, through a proprietary zoning analysis method [11] the images are converted into a compressed high-quality PDF that also embeds the recognized text.

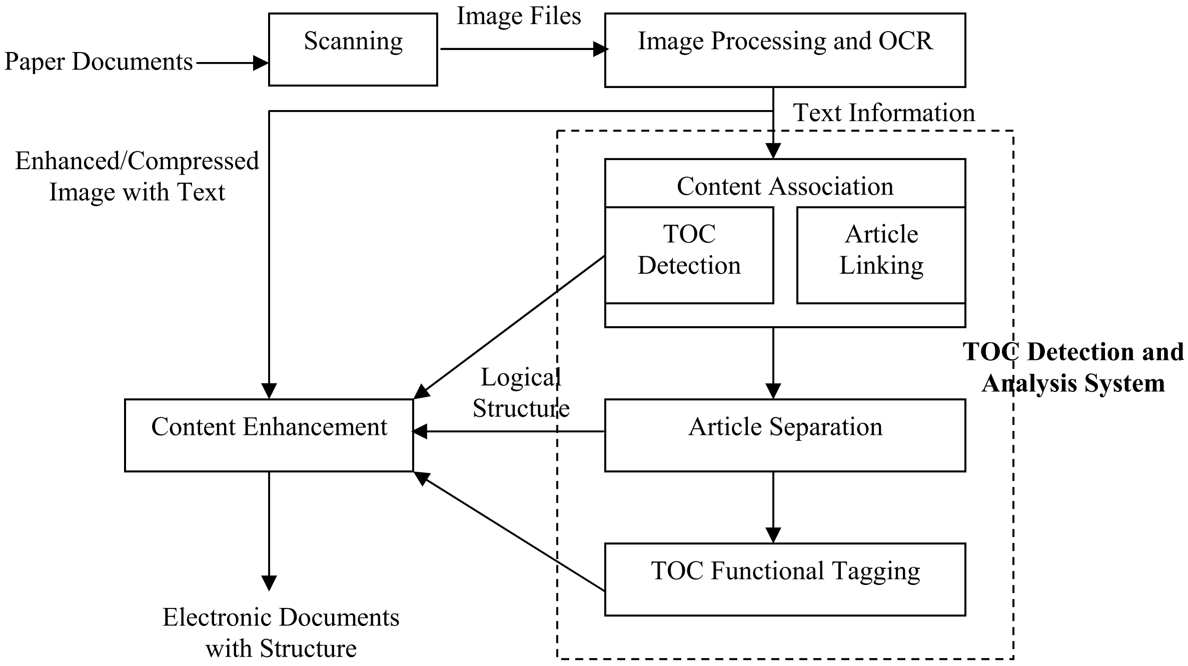


Figure 2: Overall workflow

The text and word bounding boxes on every page are passed into TOCDAS, which accomplishes several tasks:

- Automatic detection of TOC pages

In this step, TOCDAS automatically selects a subset of the pages in a document as the TOC pages. Although much research [20][21] has been done on general table detection, there is little published work on TOC detection. Among the papers we have surveyed on TOC analysis, only Mandal et al. and Luo et al. proposed layout-based methods to detect TOC pages [6][9]. The other systems either assumed that the TOC was located on fixed pages for a certain type of publication, or had the TOC pages manually selected. Because our system is designed to process documents of different categories in an unattended mode, it has to locate the TOC pages by itself. In TOCDAS, this task is carried out as part of content association algorithm.

- Article linking

TOCDAS detects the title pages and links words on the TOC pages to the title pages. In previous work [1][2], article linking relies largely on the page numbers shown on the TOC pages. The other words are then linked to articles based on geometrical proximity to the page numbers. TOCDAS does article linking separately for general text and page numbers using the content association algorithm and then combines the results.

- Article segmentation

In this step, TOCDAS segments TOC pages into a number of article references, each of which contains information associated with one article. Previous article segmentation methods utilized either layout [2] [5][6][7] or POS tagging [1]. TOCDAS takes a different approach. It refines the article linking results by grouping words linked to the same article without POS tagging or top-down layout models.

- TOC functional tagging

Each article reference obtained in article segmentation is decomposed of logical elements, such as the author(s), start page number, and title. The major challenge is to distinguish the author(s) from the title because both are non-numerical text strings. As Belaïd first proposed in [1], we use POS tagging in this step. But as a novel feature, our TOC functional tagging algorithm also takes into account layout.

The results of TOCDAS can then be used to enhance the produced electronic documents. For example, our system can embed the document structure into PDF, split a big PDF into smaller files, and help with the scanning quality assurance.

3 Content association

The most unique characteristic of the proposed method lies in content association, through which TOC detection and a significant part of TOC analysis are conducted simultaneously, helping to verify each other. As shown in Figure 3, a set of candidate TOC pages are selected. The more candidate TOC pages are included, the less likely it is that some true TOC pages will be overlooked by the system, and the more computation will be involved. We have adopted an adaptive strategy. In the initial phase, the first twenty pages are chosen as TOC candidates. If the last few candidate pages are confirmed to be real TOC pages, the candidate set will be expanded by the next twenty pages. The proposed algorithm allows TOC pages not to be one after another, but have body pages in between. Actually, the back cover pages also serving as TOC pages in journals such as *Pattern Recognition* and *IEEE PARMI*.

Then each candidate TOC page is examined separately. The contents on TOC pages are associated with those on the body pages in two ways. First, general text information about an article such as the article title, chapter/section name, and author name(s) will be present on both the TOC page and the title page of the article. Second, the start page number of an article will be listed on the TOC page. Due to the distinct natures of the two kinds of information, two algorithms, general text mining and page number matching, are designed to handle them separately. A confidence score is then assigned to the candidate

TOC page by adding together the general text mining score and page number matching score. If a candidate's score is above a certain threshold, it will be marked as a real TOC page and the body pages linked to it will be marked as title pages.

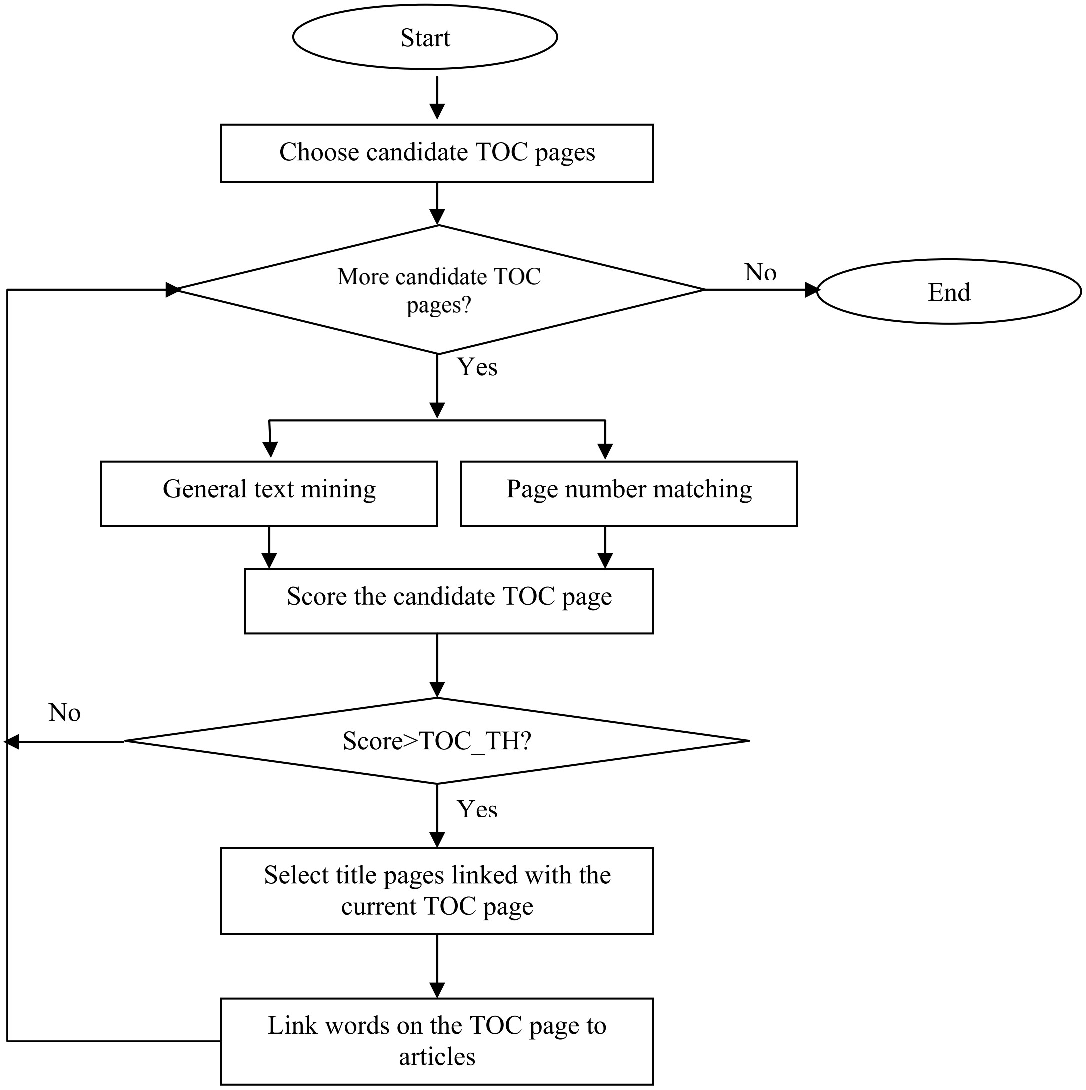

Start
Choose candidate TOC pages
More candidate TOC pages?
No
End
Yes
General text mining
Page number matching
Score the candidate TOC page
No
Score>TOC_TH?
Yes
Select title pages linked with the current TOC page
Link words on the TOC page to articles

Figure 3: Flowchart of the content association algorithm

Although the algorithm requires a page be only one type: TOC page or normal body page, the TOC pages can contain other miscellaneous information besides article references, such as the copyright and subscription information shown at the bottom of Figure 1(b).

3.1 General text mining

Although the idea of associating the text on the TOC pages with the text on the body pages sounds straightforward, it is a very hard technical problem accompanied by several challenges:

- How to retrieve only the "valid" matches?

Only matches on article titles and author names shown on the TOC and the title pages are of interest to us. In reality, however, there will be many undesired “noise” matches. Some common words such as articles (“a”, “an”, “the”, etc.) and pronouns (“it”, “we”, “you”, etc.) will appear on many pages. Catching them on both the TOC pages and the body pages does not benefit TOC analysis at all. Even meaningful texts such as article titles can appear in contexts other than the TOC pages or title pages. For example, article titles and author names can also appear as page headers or footers on every page of an article. These matches can potentially prevent us from linking the right title pages to the TOC.

- How to search efficiently?

We do not know in advance what phrases or sentences are the targets and where the matches start. The only *a priori* knowledge is that some phrases or sentences on the TOC pages will repeat on the title pages. The basic operation of the algorithm is to find the longest common sub-sequence between two long sequences. This problem has some relationship with gene sequencing [14], in which DNA fragments are concatenated together based on the sub-fragments shared among the fragments. Efficiency is critical when applying substring matching to information retrieval and document processing. For example, Zamir and Etizioni took advantage of suffix tree for efficient Web document clustering [13]. Similarly, we have to carefully design our TOC analysis algorithm to make sure that the computational complexity is acceptable on commodity personal computers.

To overcome these challenges, the following general text mining algorithm has been designed:

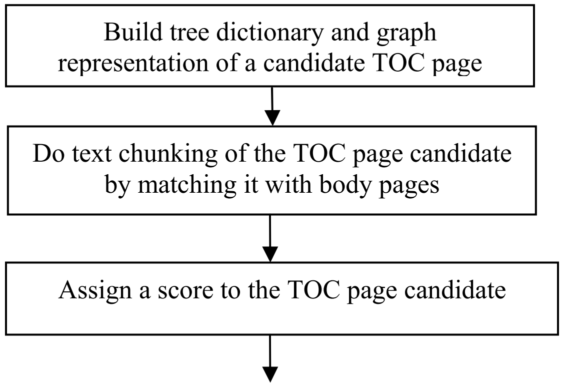

Build tree dictionary and graph representation of a candidate TOC page
Do text chunking of the TOC page candidate by matching it with body pages
Assign a score to the TOC page candidate

Figure 4: Major steps of the general text mining

3.1.1 Construct a tree-structured dictionary for each candidate TOC page

In order to accelerate the string matching, a dynamic tree-structured dictionary is created for each candidate content page. Once such a dictionary (see Figure 5) is built, it takes much less time to find out if a word is on the candidate TOC page through nonlinear tree search than through sequential linear search. The experiment has shown a three-fold speedup using this technique.

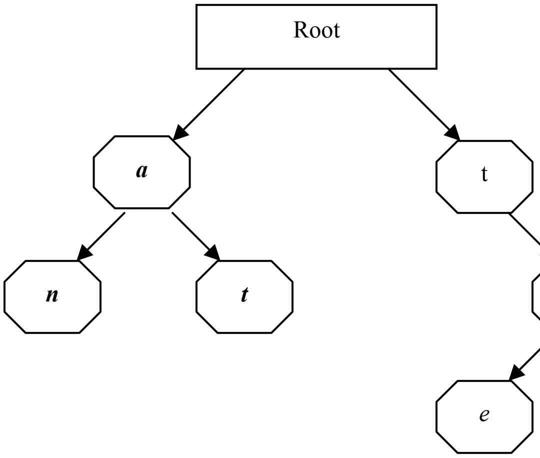


Figure 5: Tree dictionary structure based on TOC Pages

(Nodes that can serve as ends of words are in bold italic font)

3.1.2 Represent the candidate TOC page as a directional graph

The next question is how to describe the candidate content pages. The simplest approach is to use a linked list to thread together all of the words. The drawback is that repeated words will appear more than once in the list. So we model the candidate content page as a directional graph with each node as a unique word. On such a graph, each node uniquely corresponds to a leaf element of the tree-structured dictionary resulting from the earlier step. The edge between two nodes has a state to indicate the reading order. Figure 6 represents the sentence “a student is a person who studies in a school.” The biggest advantage of this graph representation is its high efficiency. For example, in order to find out all the phrases starting with the word “a”, we first look up the vertex in the tree dictionary. Then we can immediately locate the phrases starting from that word (“a student”, “a school”, and “a person”) by traversing all of the edges from that vertex. Additionally, if both the start state and the end state are given, one phrase can be uniquely decided. We define such a pair (start state, end state) as a *Range*. For example, as shown in Figure 6 Range (4,6) corresponds to the phrase “person who studies”.

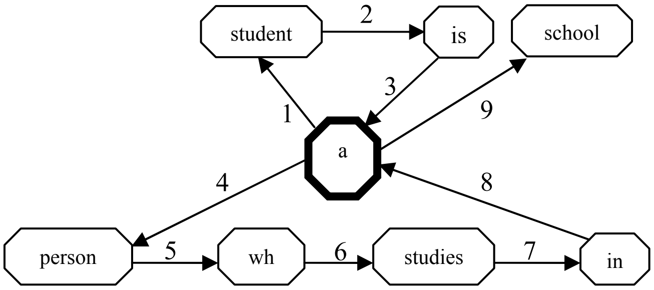


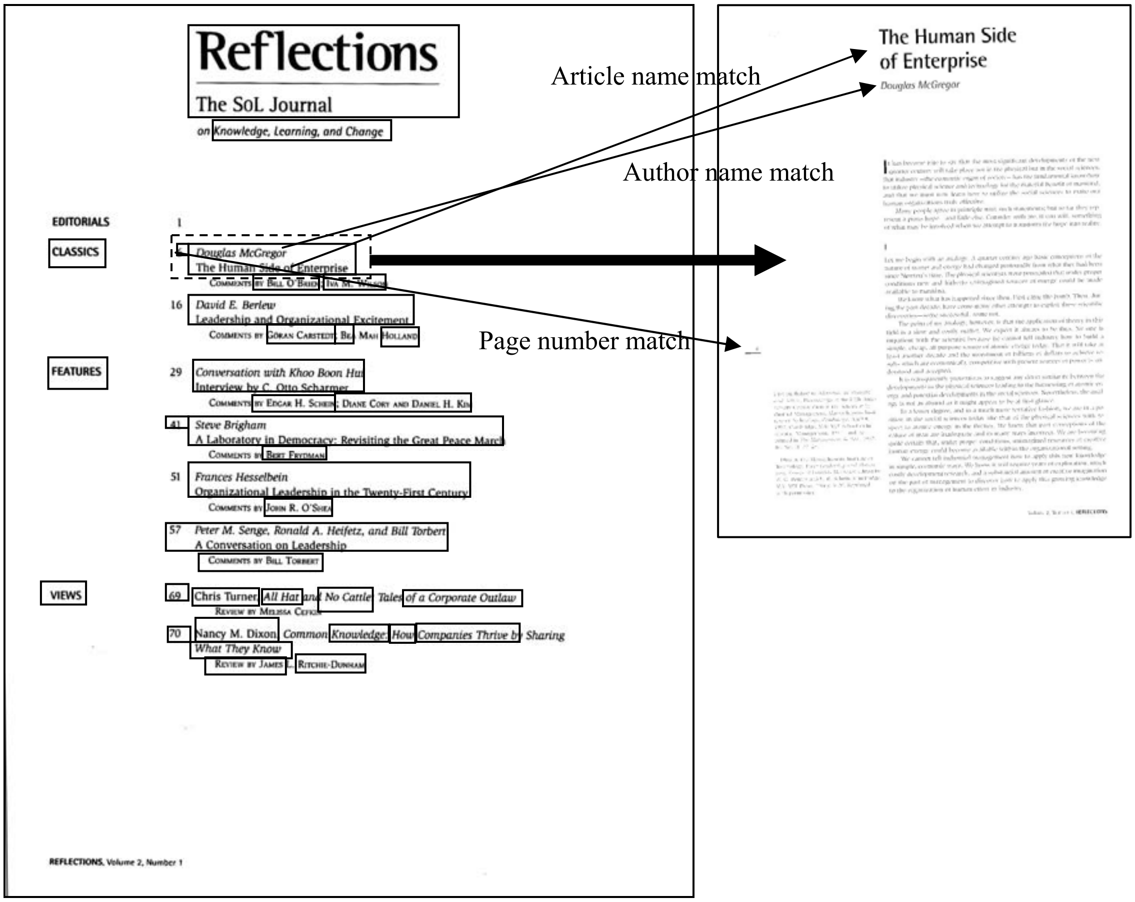
Figure 6: Graph description of TOC pages

3.1.3 Divide the candidate TOC page into text chunks

Many text chunks in the TOC page should have good matches in the title pages. The following algorithm is used to detect the matched Ranges:

```
RangeList:= {}
foreach <Word>∈ BodyPages
    Find the longest match starting with Word between BodyPages and TOC. This match is put into a Range (s,e).
    If Range (s,e) overlaps with an existing Range in RangeList, the two Ranges are merged. Otherwise it is inserted into the RangeList.
endfor
```

Figure 7 displays an example of text chunking. The words in the same rectangle are put together by text chunking. It can be seen that most article titles and author names are properly grouped. Some phrases fall outside of the groups due to OCR errors. It should be emphasized that the grouping result comes exclusively from the above text chunking process without using geometric layout information.



Reflections

The SoL Journal

on Knowledge, Learning, and Change

EDITORIALS 1

CLASSICS 5 Douglas McGregor
The Human Side of Enterprise
Comments by Bill O'Brien; Iva M. Wilson

16 David E. Berlew
Leadership and Organizational Excitement
Comments by Göran Carstedt; Bea Mah Holland

FEATURES 29 Conversation with Khoo Boon Hui
Interview by C. Otto Scharmer
Comments by Edgar H. Schein; Diane Cory and Daniel H. Kim

41 Steve Brigham
A Laboratory in Democracy: Revisiting the Great Peace March
Comments by Bert Frydman

51 Frances Hesselbein
Organizational Leadership in the Twenty-First Century
Comments by John R. O'Shea

57 Peter M. Senge, Ronald A. Heifetz, and Bill Torbert
A Conversation on Leadership
Comments by Bill Torbert

VIEWS 69 Chris Turner, All Hat and No Cattle: Tales of a Corporate Outlaw
Review by Melissa Cefkin

70 Nancy M. Dixon, Common Knowledge: How Companies Thrive by Sharing What They Know
Review by James L. Ritchie-Dunham

REFLECTIONS, Volume 2, Number 1

The Human Side
of Enterprise

Douglas McGregor

Figure 7: Article linking result on a TOC page

(Words inside the dashed box are linked with a title page shown on the right side)

3.1.4 Score the candidate TOC page based on text mining

As an initial result from the above steps, it is possible that one Range will be linked to multiple body pages. Of course, we should keep only one link for each Range. So "winner-take-all" is employed in the evaluation stage. All of the links to a Range are scored based on several factors, including the match length, the ordering of body pages, the font size, etc. The best link is kept and the other links are deleted. The link scores on a candidate TOC page will be summed up as this page's general text mining score.

3.2 Page number matching

Page number matching also contributes to TOC detection and article linking, especially when the text contains OCR errors or general text mining cannot provide enough evidence (for example, in books, the chapter titles sometimes can be very short and usually there are no author names for individual chapters).

3.2.1 Score TOC page candidates based on page numbers

Page numbers have several distinct characteristics: They are roman numerical strings, usually located at the start or end of each article reference, vertically aligned, and incremental from column to column, from line to line, and from page to page. Thus, the numerical strings located at the beginning or end of lines or vertically aligned are extracted as page number candidates. Through depth-first search, biggest incremental subset of the page number candidates is selected. The page number candidates that are not in this incremental subset are removed. Then a candidate TOC page is assigned a confidence score based on the quantity (measured as the number of detected page numbers) and the quality (measured in terms of the incremental pattern, alignment, etc.) of detected page numbers. The sum of this score and the candidate's general text mining score (described in Section 3.1) is used to decide if this candidate is really a TOC page or not.

3.2.2 Link page numbers to articles

If the candidate TOC page is confirmed, the page numbers on that TOC page are linked to individual articles. There are two types of page numbers. The page numbers printed on the TOC pages are the logical page numbers (LPNs), which are different from the physical page numbers (PPNs). PPNs are imposed by the scanning process. For example, if all pages in a book are scanned, the front cover page will be the first physical page. Since it is the PPN that can uniquely identify a page, we have to convert LPNs to PPNs in order to map LPNs to individual articles. Although this conversion depends on the LPNs printed on the body pages, it is not trivial for several reasons. First, many journals do not actually have the page numbers printed on the article title pages. Second, the locations of page numbers on body pages are very different from journal to journal. Third, page numbers on body pages are typically in small font sizes and are more likely to be misrecognized by OCR software than other text. Consequently, solely relying on the page

number of a single page can be very fragile. We have designed a more robust weighted histogram filter method, which first decides the offset between the LPN and PPN through statistics across multiple pages and then infers the LPN from the PPN and the offset:
Step 1: Extract a set of LPN candidates on Page n:

$$S(n) = \{s(n,i) \mid i = 1, \ldots I(n)\} \qquad \textbf{(1)}$$

where N is total number of pages, $1 \le n \le N$ *, and I(n) is the size of S(n).*
Given a page, the LPN candidates are the words that are numerical strings and located in the border area of a page (for example, the top line, the bottom line, the first and the last word of each line). This step tends to be very inclusive so that some candidates will be non-page-number numerical strings happening to be located in the area of interest.

Step 2: Calculate the set of differences between the candidate LPNs and the PPN:

$$D(n) = \{d(n,i) \mid i = 1, \ldots I(n)\} \qquad \textbf{(2)}$$

where d(n,i)=s(n,i)-n.
Step 3: Run the weighted histogram filter operation to reduce the influence of noise LPN candidates.
For Page n, a weighted histogram of d(x, i) is calculated for a window that spans the w preceding pages, Page n, and w following pages. The weight for each page in the window is assigned as:

$$c(x,n) = \begin{cases} 2\ (if\ x = n) \\ 1\ (if\ -w + n \le x \le w + n\ and\ x \ne n) \end{cases} \qquad \textbf{(3)}$$

The accumulated strength of offset value *diff* is calculated as:

$$strength(diff, n) = \sum_{x=-w+n}^{x \le w+n} c(x,n) * count(diff, n)$$
$$where\ count(diff, n) = \begin{cases} 1\ (if\ diff \in D(x) \\ 0\ (otherwise) \end{cases} \qquad \textbf{(4)}$$

Then the *diff* with the maximal strength, denoted as *DIFF(n)*, is selected, and the LPN is inferred from *DIFF(n)* and n:

$$DIFF(n) = \arg\max_{diff}(strength(diff, n))$$
$$LPN(n) = n + DIFF(n) \qquad \textbf{(5)}$$

It is worth explaining why we calculate an offset for each page instead of a common offset for the whole document. If all of the pages in a book or journal are sequentially scanned without any omissions or mistakes, the offset will be the same for every page. However, in the real-world operations, it is inevitable that some pages will be skipped either intentionally or by mistake. For example, the operator may choose to skip blank pages to increase efficiency. In such situations, the offset is no longer a constant throughout the whole document. The choice of window width w reflects the tradeoff between the statistical accuracy and the frequency of local variations. If w is large, more pages will fall into the window and the statistics can be more accurate, but the averaging effect may overwhelm real local variations of the offsets. In our experimental results, when w is set to 6, this algorithm works very well on a wide variety of body page layouts, in the absence of page numbers on title pages, or with moderate degree of OCR errors.

With the LPN to PPN conversion, each page number x on the confirmed TOC page is linked to Page PPN(x).

3.3 Selection of title pages and links

Based on each confirmed TOC page, a non-TOC page P is assigned a title-page confidence score that is the sum of the general text mining score and the page number matching score, and the maximum is P’s final title-page score.

$$\begin{aligned} & Title\_page\_score(P) = \max_{i \le T}(Page\_number\_score(P, TOC_i) \\ & + Text\_score(P, TOC_i)) \end{aligned} \quad \textbf{(6)}$$

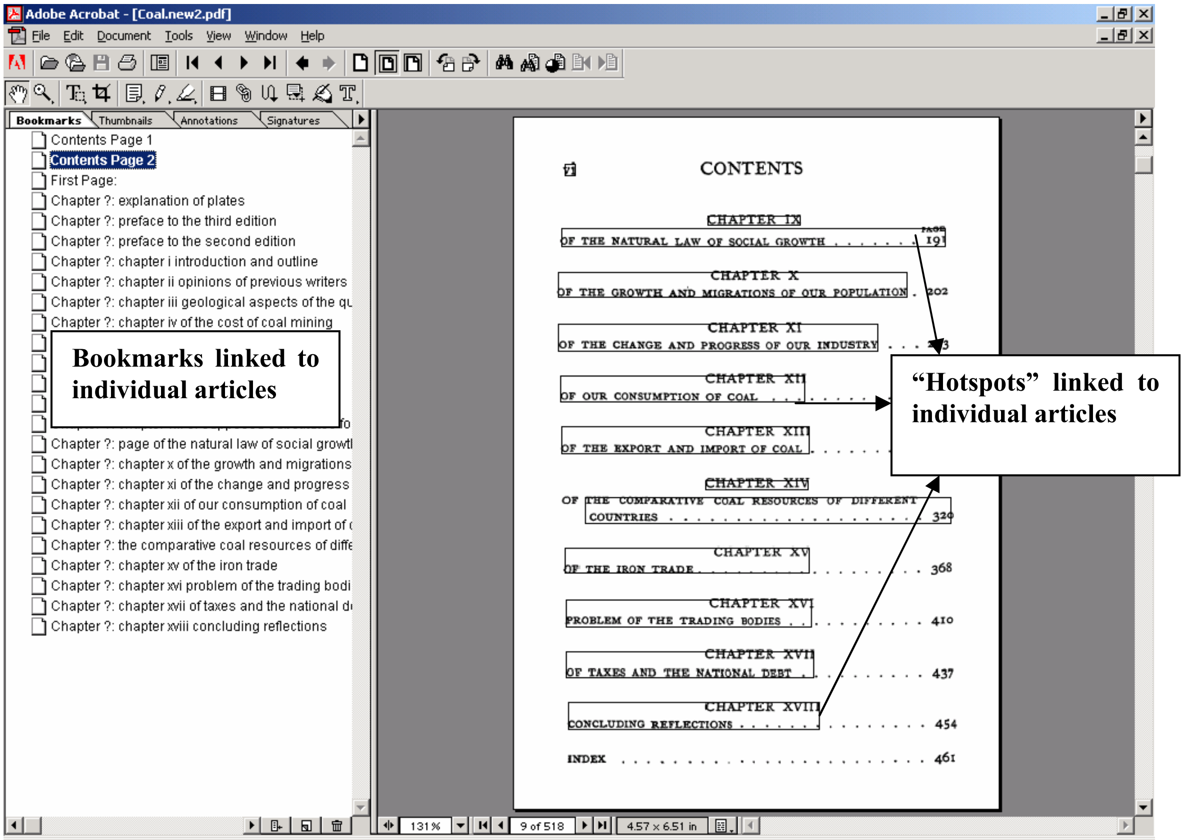
where T is the number of confirmed TOC pages.

If *Title_page_score(P)* is above a certain threshold, P will be confirmed as a title page. After the title pages are decided, all links established in general text mining and page number matching are revisited. Only those that connect a confirmed TOC page and a confirmed title page are kept.

3.4 Applications of content association

We have explored several useful applications built on top of the core content association algorithms: document navigation [18], journal splitting [4], and scanning quality assurance.

Once the TOC pages are detected and the links to titles pages are identified, navigation capabilities can be embedded into electronic documents. The navigation is added to the PDF files in two ways. First, a bookmark is automatically generated for each article (left panel of Figure 8). The text description of a bookmark comes from the Range(s) pointing to an article. When the user clicks on a bookmark, the right panel will display the linked article. Second, the Ranges on the TOC pages are marked out and linked to individual articles (see the right panel of Figure 8). When the user clicks on a text region on the TOC page, the PDF viewer will display the linked title page.


Adobe Acrobat - [Coal.new2.pdf]
File Edit Document Tools View Window Help
Bookmarks Thumbnails Annotations Signatures
131%
9 of 518
4.57 x 6.51 in

Contents Page 1
Contents Page 2
First Page:
Chapter ?: explanation of plates
Chapter ?: preface to the third edition
Chapter ?: preface to the second edition
Chapter ?: chapter i introduction and outline
Chapter ?: chapter ii opinions of previous writers
Chapter ?: chapter iii geological aspects of the qu
Chapter ?: chapter iv of the cost of coal mining
Chapter ?: page of the natural law of social growtl
Chapter ?: chapter x of the growth and migrations
Chapter ?: chapter xi of the change and progress
Chapter ?: chapter xii of our consumption of coal
Chapter ?: chapter xiii of the export and import of
Chapter ?: the comparative coal resources of diffe
Chapter ?: chapter xv of the iron trade
Chapter ?: chapter xvi problem of the trading bodi
Chapter ?: chapter xvii of taxes and the national d
Chapter ?: chapter xviii concluding reflections

Bookmarks linked to individual articles

vi CONTENTS

CHAPTER IX
OF THE NATURAL LAW OF SOCIAL GROWTH . . . . PAGE 191

CHAPTER X
OF THE GROWTH AND MIGRATIONS OF OUR POPULATION . 202

CHAPTER XI
OF THE CHANGE AND PROGRESS OF OUR INDUSTRY . . . . 2[illegible]3

CHAPTER XII
OF OUR CONSUMPTION OF COAL . . . .

CHAPTER XIII
OF THE EXPORT AND IMPORT OF COAL . . . .

CHAPTER XIV
OF THE COMPARATIVE COAL RESOURCES OF DIFFERENT COUNTRIES 320

CHAPTER XV
OF THE IRON TRADE 368

CHAPTER XVI
PROBLEM OF THE TRADING BODIES 410

CHAPTER XVII
OF TAXES AND THE NATIONAL DEBT 437

CHAPTER XVIII
CONCLUDING REFLECTIONS 454

INDEX 461

"Hotspots" linked to individual articles

Figure 8: Two ways of navigation in PDF files

After identifying the TOC pages and the title pages, it is straightforward to split the whole document into smaller files, each of which contains an article or the TOC. The PDF file for a whole book/journal can have a size ranging from several megabytes to tens of megabytes. It can take a long time to download the file through low-speed Internet access, such as the dial-up service. Accordingly, if the book/journal is already split into smaller units such as individual articles, the user can simply download the contents he/she is interested in. In addition, by splitting a book/journal, the content provider can increase sales by selling the contents in finer granularity. The content association algorithm has been used to split 256 books in MIT Press' Classic Series into individual chapters, which are available from MIT's Cognet website, a brain science online community [23].

The PPN to LPN mapping introduced in Section 3.2 can benefit scanning quality assurance (QA). The program automatically pinpoints the pages where the offset between PPN and LNP changes. A human operator can then inspect those pages to make sure that no pages have been left out by mistake. This tool has been incorporated into our re-mastering system's QA process.

3.5 Experimental results

The content association algorithm has been tested on journals and books from MIT Press. Table 1 shows the statistics of the testing samples. Since content association accomplishes two tasks, TOC page detection and article linking, we measure them separately. TOC page detection can be evaluated by the detection error rate (the sum of deletion and insertion rate). Article linking can be measured in a number of ways. We can count the percentage of words on the TOC pages correctly linked to title pages. However, this measurement requires a considerable amount of manual ground truthing and is sometimes ambiguous. For example, it is possible that only part of the article name or the author names is linked to a title page because of OCR errors or even the documents themselves. On the other hand, when title pages are incorrectly inserted or deleted, they will definitely have big impact on the downstream applications, such as document navigation or document splitting. So we use the title page detection error rate (the sum of deletion and insertion rate) as the metric of article linking. Table 2 shows the testing results. We can see that all of the TOC pages are correctly detected and the title page detection error rate is 2%. Besides, if we turn off general text mining and only use page number matching, the title page detection error rate would increase to 3%. This demonstrates the benefit of combining general text mining and page number matching approaches.

It is worth mentioning two facts about the testing results. First, the test is a strictly open test. None of documents we have seen in the algorithm design stage have been used in testing. Second, the second column of Table 1 shows that the documents come from nice journal titles. As can be seen in Figure 1, different journal titles can have completely different layout styles. These two facts strongly support our opening statement: The content association method can be directly applied to a wide range of documents without the need to build or learn the models for individual documents.

Table 1: Testing set

Number of documents	Number of different journal titles	Number of total pages	Number of TOC pages	Number of articles
151	9	26,487	282	3,967

Table 2: Experimental results on TOC page and title page detection

TOC page detection error rate	Title page detection error rate	Title page detection error rate using only page number matching
0	2.0%	3.0%

The following table shows the computation time for one book and one journal. The computer is a 733 MHz Pentium III workstation running Windows 2000. The time spent on content association is only a fraction of the time spent on the re-mastering process of converting TIFF files to PDF files. The re-mastering of the 518-page book takes about 15 minutes, and the re-mastering of the journal takes about 5 minutes.

Table 3: Computation times

No	Category	No of Pages	No of Words	No of TOC Pages	Sample Page	Time (seconds)

1	Book	518	115,000	2	Figure 8	118
2	Journal	168	126,000	1	Figure 7	49

We have also investigated how the quality of the OCR engines affects the linking algorithm. The results based on two commercial OCR engines (see [17] for more details on the engines) are compared. Engine A has a character error rate of 0.46% while Engine B has an error rate of 1.1%. More words on the TOC page are linked to articles by using Engine A. In addition, some articles that can be detected with Engine A are missed when Engine B is used. In conclusion, a superior OCR engine can still help the content association algorithm, although the proposed algorithm is already not very sensitive to OCR errors.

4 Article separation

Figure 9 displays the hierarchical logic structure of a TOC. On the top level, a TOC is composed of a number article references. Each article reference consists of several logical elements: author (this element may be absent in books), title, and page number. Words are the most basic objects. Word groups sit between articles and words. They are collections of words belonging to the same line and the same logical element. Through the content association process, many words on the TOC pages have already been linked to individual articles. Guided by the word-article linkages, the article separation algorithm works in a bottom-up manner. It first lines up words into word groups and then clusters word groups into article reference blocks. This bottom-up approach does not assume any TOC page layout templates or rules, which are generally required by many existing top-down article separation methods and can limit the applicability of those methods. On the other hand, it is also different from common bottom-up layout analysis algorithms by fully leveraging the word-article relationship obtained in content association.

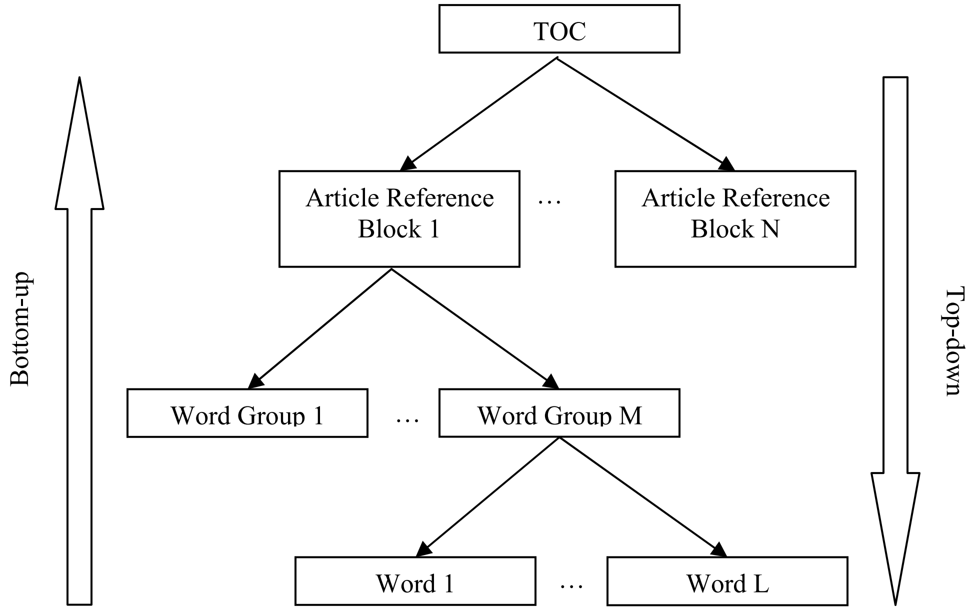


Figure 9: Two approaches to TOC analysis

4.1 Word group generation

The first step in article separation is to generate word groups. Each physical line is traversed from left to right. If the gap between the current word and the previous one is larger than 1.5 times of the median word gap of this page, a new word group is created to include the current word. Otherwise, the current word is added to current word group. The article ID (specified by the page number of the title page) of a word group is the dominant article ID label of its constituent words.

4.2 Clustering word groups into article reference blocks

Under the assumption that no two articles will start on the same page, the word groups are then clustered into blocks based on geometrical proximity and the article IDs of the word groups. The following sequential merge algorithm is introduced:

Step 1: Find an unexpanded word group $g_x = g_0$ (the seed) and create an article reference block b_i. Stop clustering if there are no more unexpanded word groups.

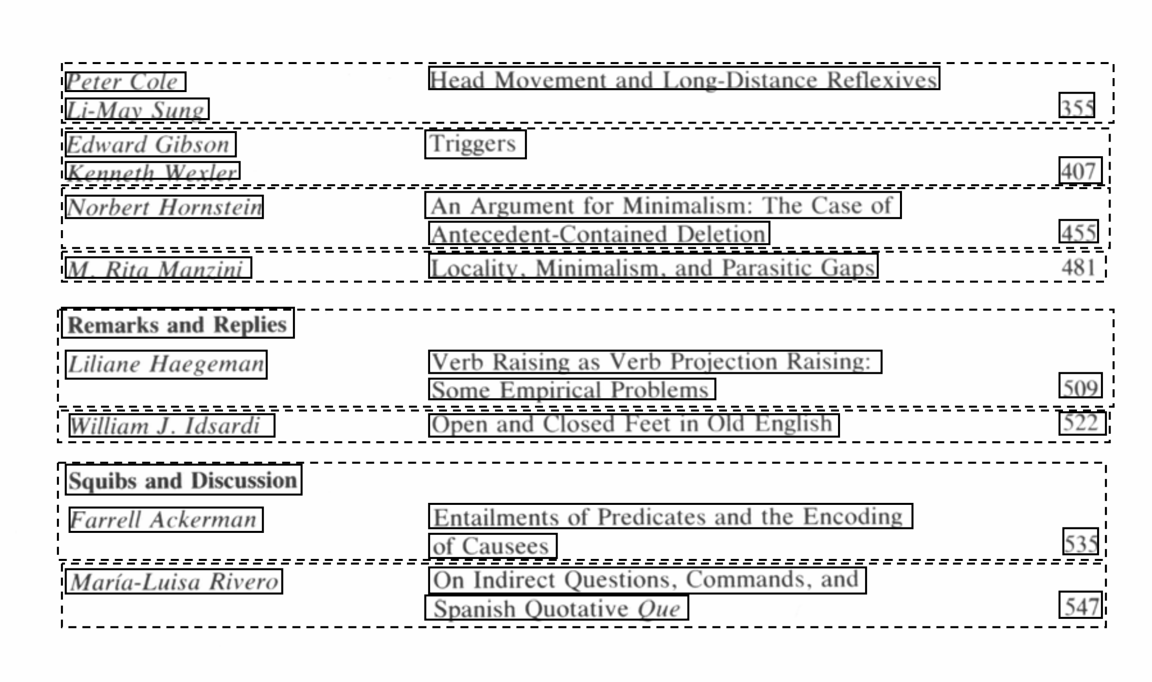
Step 2: Mark current word group g_x as expanded and add it to b_i.

Step 3: Evaluate all unexpanded word-groups g_j in the neighborhood. Any g_j that is geometrically close enough to g_x and shares the same article ID labels as g_x is selected.

Step 4: Every g_j selected in Step 3 becomes the current word group g_x, and go back to Step 2.

Step 5: Increase i by 1 and go back to Step 1.

Figure 10 shows the word groups and article reference blocks generated by the article separation algorithm on one TOC page. The statistical result will be presented together with TOC functional tagging in Section 5.



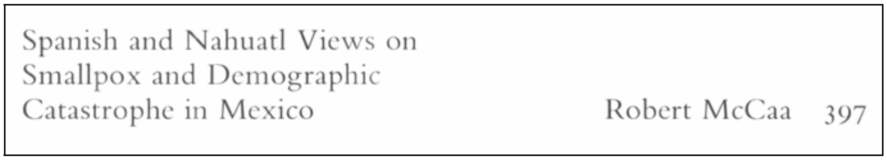
Peter Cole *Li-May Sung*	Head Movement and Long-Distance Reflexives	355
Edward Gibson *Kenneth Wexler*	Triggers	407
Norbert Hornstein	An Argument for Minimalism: The Case of Antecedent-Contained Deletion	455
M. Rita Manzini	Locality, Minimalism, and Parasitic Gaps	481
Remarks and Replies		
Liliane Haegeman	Verb Raising as Verb Projection Raising: Some Empirical Problems	509
William J. Idsardi	Open and Closed Feet in Old English	522
Squibs and Discussion		
Farrell Ackerman	Entailments of Predicates and the Encoding of Causees	535
María-Luisa Rivero	On Indirect Questions, Commands, and Spanish Quotative *Que*	547

Figure 10: An example showing the word groups (rectangles in solid lines) and article reference blocks (rectangles in dashed lines) generated from clustering

5 TOC functional tagging

Functional tagging divides an article reference block, as identified by the article separation algorithm, into semantic elements: author names, articles, and page numbers. Although POS tagging is very useful for this purpose [1], it is still a great challenge to design a TOC functional tagging algorithm that can work reasonably well across a wide range of documents. First, author names and titles sometimes cannot be differentiated from each other by POS tagging alone. Second, POS tagging itself is still an open problem and will make mistakes. Third, OCR errors will adversely affect POS tagging [15]. On the other hand, the physical layout of a document provides another imperfect source for TOC functional tagging. Thus, we have introduced a method that combines POS tagging and layout.

As discussed in the previous section, each article reference block consists of a set of word groups. For example, the article reference block in Figure 11 consists of five word groups shown in Figure 12. Each word group has four possible functional tags: section (s), title (t), author (a), or page number (n). So this block can potentially have 4^5 different tagging combinations, such as (t, t, t, a, n), (a, t, t, a, n), … (a, a, a, a, n). Certain linguistic rules can be applied to reduce the search space. For example, numerical strings should never be tagged as an author name. The remaining combinations are then evaluated in terms of both POS tagging and layout analysis. Each tagging combination is assigned a total cost that is the sum of the linguistic cost and the layout cost. The tagging combination with the least cost is then selected as the final result.



Spanish and Nahuatl Views on
Smallpox and Demographic
Catastrophe in Mexico Robert McCaa 397

Figure 11: An example of TOC segment

g1:	Spanish and Nahuatl Views on
g2:	Smallpox and Demographic
g3:	Catastrophe in Mexico
g4:	Robert McCaa
g5:	397

Figure 12: Word groups corresponding to Figure 11

We use Cogilex QuickTag [16] to extract POS tags. Figure 13 shows the POS tagging results of Figure 12. Then the linguistic costs for different logical elements are assigned based on the factors listed in Table 4.

g1:	CN	CC	PN	CN	PREP
g2:	CN	CC	PN		
g3:	CN	PREP	PN		
g4:	PN	PN			
g5:	NUM				

Figure 13: POS tagging results of Figure 12 (CN = common noun; PN = proper noun; CC = conjunction; PREP = preposition; NUM = numeral; ADJ = adjective; UN = unknown)

Table 4: Factors affecting the linguistic costs for different logical elements

Logical elements	**Factors affecting the linguistic costs**
Author (a)	The percentage of proper nouns
Page number (n)	The number of characters in the word group and the presence of numerals
Section (s)	The relative position in a block, the number of words, and the presence of certain words
Title (t)	Inversely affected by the costs of being an author element, a page number element, or a section element

The layout cost measures closeness among word groups in the same logical element. A straightforward measurement is to calculate the direct distance between any two groups. However, this direct distance does not reflect the hidden connections that play an important role in human judgment. In Figure 11, although the direct distance between Group 1 and Group 3 is not small, the two groups do not look far away from each other due to the bridge effect of Group 2. Based on this observation, we have adopted Floyd's all-pairs shortest-path algorithm [19], which calculates the shortest distance between every pair of points in a graph. We first extract the distance matrix:

$$D = [d_{ij}] \text{, where } N \text{ is the number of word groups and } 0 \le i, j < N \qquad \textbf{(7)}$$

d_{ij} measures the direct distance between g_i and g_j. If the two word groups are on the same line, the distance is determined by their horizontal distance. If they are on the neighboring two lines, the distance is determined by the degree of their horizontal overlap. Using the all-pairs shortest-path algorithm, the raw distance matrix *D* is then converted to the shortest-path matrix *S*, which measures the connectivity among all word groups in the block. The key difference between *S* and *D* is that *S* takes into account the indirect path between two points in order to find the global shortest path. Then the layout cost from this title region is calculated as the sum of the elements in *S*. Similarly, the costs for the author element, page number element, and section element are also calculated. The total layout cost for a block is a sum of them.

We have tested the proposed article separation and functional tagging methods on 19 journals from MIT Press. They cover 10 different journal titles and consist of 306 articles. The accuracy for article separation is 94.0%. The tagging accuracy is 100% for page number elements, 94.4% for title elements, and 90.5% for author elements. The accuracy is defined as the percentage of correctly identified article references or logical elements. We have not counted the accuracy for section elements since they offer little value for cataloging purpose. Most confusion between title elements and author elements is caused by OCR errors and non-English author names, which can generate out-of-vocabulary words and result in POS tagging errors.

6 Conclusions

After a comprehensive survey of existing work on TOC detection and analysis, we have pinpointed the fundamental characteristic distinguishing TOCs from ordinary tables: The contents in the TOC are literally associated with those on the body pages. Taking advantage of this characteristic, we have designed and implemented a complete TOC processing solution. Through content association, document-specific models are avoided and the system can thus directly process a wide range of documents without tuning or training. The results of content association also provide guidance to later processing steps. For example, the article separation algorithm incorporates the word-article linkage knowledge into the bottom-up clustering. In addition, significant effort has been devoted to making the system more robust against OCR errors and other variations that are inevitable in a serious high-volume digital library project. First, information fusion is widely employed. Content association combines general text mining and page number matching, and TOC functional tagging considers both the linguistic cost and the layout cost. Second, the method is mostly statistical rather than rule-based. In intermediate steps, quantitative scoring instead of hard decision is used as much as possible. We have also enumerated several interesting applications, such as automatically embedding navigation capabilities into re-mastered documents and scanning quality assurance.

In retrospect, the content association algorithm bears similarity to a seemingly unrelated but hot topic: Web link mining. Most modern Internet search engines, including Google, take advantage of the fact that the authoritative Web pages are the ones frequently pointed to by other pages. In this paper, the TOC pages are the hub pages referenced by other body pages. However, there is an important difference between this work and Web link mining. The linking information is explicitly marked with special HTML tags in Web link mining, while the links are only implied in the form of repeated sentences or phrases in the context of TOC analysis. In order to discover those hidden links, we have designed an efficient algorithm based on dynamic tree dictionary, TOC graph representation, and text chunking.

One interesting topic for future research is the processing of non-English or multi-lingual TOC pages. The basic operation of content association is text string matching, which works on all languages. The article separation algorithm operates on top of content association and physical layout, and thus it is independent of the languages as well. However, TOC functional tagging utilizes on the language-dependent POS tagging and currently only supports English. We plan to add support of other languages.

Acknowledgements

We would like to thank Sheelagh Hudleston and Steven Simske for many valuable discussions and Sherif Yacoub for his help on the system integration. We are also grateful to John Burns and Teresa Ehling, who have championed this research from the start.

References

[1] A. Belaïd, "Recognition of Table of Contents for Electronic Library Consulting," *International Journal on Document Analysis and Recognition*, vol. 4 no. 1, pp. 35-45, 2001

[2] C. Lin, Y. Niwa, and S. Narita, "Logical Structure Analysis of Book Document Images Using Contents Information," *Proc. of 4th International Conference on Document Analysis and Recognition*, pp. 1048-1054, Ulm, Germany, August 1997

[3] S. Satoh, A. Takasu, and E. Katsura, "An Automated Generation of Electronic Library Based on Document Image Understanding," *Proc. of 3rd International Conference on Document Analysis and Recognition*, pp. 163-166, Tokyo, Japan, August 1995

[4] X. Lin, "Text-mining Based Journal Splitting," *Proc. of 7th International Conference on Document Analysis and Recognition*, pp. 1075-1079, Edinburgh, UK, August 2003

[5] G. A. Story, L. O'Gorman, D. Fox, L. L. Schaper, H. V. Jagadish, "RightPages Image-Based Electronic Library for Browsing and Alerting," *IEEE Computer*, pp. 17-25, September 1992

[6] S. Mandal, and S. P. Chowdhury, "Automated Detection and Segmentation of Table of Contents Page from Document Images," *Proc. of 7th International Conference on Document Analysis and Recognition*, pp. 398-402, Edinburgh, UK, August 2003

[7] S. Tsuruoka, C. Hirano, T. Yoshikawa, and T. Shinogi, "Image-based Structure Analysis for a Table of Contents and Conversion to XML Documents," *Workshop on Document Layout Interpretation and Its Application (DLIA 2001),* Seattle, USA, September 9, 2001

[8] F. Le Bourgeois, H. Emptoz, and S. Souafi Bensafi, "Document Understanding Using Probabilistic Relaxation: Application on Tables of Contents of Periodicals," *Proc. of 6th International Conference on Document Analysis and Recognition*, pp. 508-512, Seattle, USA, September 2001

[9] Q. Luo, T. Watanabe, and T. Nakayama, "Identifying Contents Page of Documents," *Proc. of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 696-700, 1996

[10] F. He, X. Ding, and L. Peng, "Hierarchical Logical Structure Extraction of Book Documents by Analyzing Tables of Contents," *Proc. SPIE Conference on Document Recognition and Retrieval IX*, pp. 6-13, San Jose, USA, Jan 2004

[11] S. Simske, and X. Lin, “Creating Digital Libraries: Content Generation and Re-Mastering,” *Proc. International Workshop on Document Image Analysis for Libraries*, pp. 33-45, Palo Alto, January 2004

[12] P. Hannappel, R. Klapsing, and G. Neumann, “MSEEC-A Multi Search Engine with Multiple Clustering,” *Proc. of the ’99 Information Resources Management Association International Conference*, Hershey, Pennsylvania, May 1999

[13] O. Zamir, and O. Etzioni, “Web Document Clustering: A Feasibility Demonstration,” *Proc. of 19th International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 46-54, 1998

[14] G. Myers, “Whole-Genome DNA Sequencing,” *IEEE Computational Engineering and Science*, pp. 33-43, May-June 1999

[15] X. Lin, “Impact of Imperfect OCR on Part-of-speech Tagging,” *Proc. of 7th International Conference on Document Analysis and Recognition*, pp. 284-288, Edinburgh, UK, August 2003

[16] Cogilex Website, http://www.cogilex.com

[17] X. Lin, “Reliable OCR Solution for Digital Content Re-mastering,” *Proc. SPIE Conference on Document Recognition and Retrieval IX*, pp. 223-231, San Jose, USA, January 2002

[18] X. Lin, and S. Simske, “Automatic Document Navigation for Digital Content Re-Mastering,” *Proc. SPIE Conference on Document Recognition and Retrieval XI*, pp.66-73, San Jose, USA, January 2004

[19] L. Allison, T. I. Dix, and C. N. Yee, “Shortest Path and Closure Algorithms for Banded Matrices,” *Information Processing Letters*, vol. 40 no. 6, pp. 317-322, 1991

[20] J. Hu, R. Kashi, D. Lopresti, and G.Wilfong, “Medium-independent Table Detection,” *Proc. SPIE Conference on Document Recognition and Retrieval VII*, pp. 291–302, San Jose, USA, January 2000

[21] Y. Wang, I. T. Phillips, and R. Haralick, “Table Detection Via Probability Optimization,” *Document Image Analysis System V, 5th International Workshop D*AS 2002, pp. 272-282, Princeton, USA, August 2002

[22] MIT Press, “Classics Book Collection Release Announcement,” http://mitpress.mit.edu/main/feature/classics/MITPClassics_release.pdf

[23] CogNet Website, http://cognet.mit.edu