Automatic Document Navigation for Digital Content Re-mastering

Xiaofan Lin, Steven Simske
Imaging Systems Laboratory
HP Laboratories Palo Alto

HPL-2003-197
September 23rd , 2003*

E-mail: {xiaofan.lin, steven.simske} @hp.com

digital content re-mastering, document structure analysis, print on demand, content linking, OCR

This paper presents a novel method of automatically adding navigation capabilities to re-mastered electronic books. We first analyze the need for a generic and robust system to automatically construct navigation links into re-mastered books. We then introduce the core algorithm based on text matching for building the links. The key features of the system are also described with a discussion of the experimental results on the MIT Press corpus.

Automatic Document Navigation for Digital Content Re-mastering

Xiaofan Lin and Steven Simske
Hewlett-Packard Laboratories, 1501 Page Mill Road, MS 1203, Palo Alto, CA 94304
Email: {xiaofan.lin, steven.simske}@hp.com

ABSTRACT

This paper presents a novel method of automatically adding navigation capabilities to re-mastered electronic books. We first analyze the need for a generic and robust system to automatically construct navigation links into re-mastered books. We then introduce the core algorithm based on text matching for building the links. The key features of the system are also described with a discussion on the experimental results.

Keywords: digital content re-mastering, document structure analysis, print on demand, content linking, OCR

1. INTRODUCTION

Digital Content Re-Mastering (DCRM) is the conversion of legacy contents into digital form and their storage for future reading and printing. One typical application of DCRM is for generating digital libraries: Paper documents (for example, books, journals, and magazines) are converted into digital form as, for example, PDF and DJVU [1] formats. The digitized material can then be distributed using CD-ROMs or the Internet. The digitized contents can also be used in print-on-demand (POD) systems. For example, in a joint project with MIT Press, we have re-mastered 1,750 out-of-print book/journal titles in the MIT Classics [2]. Due to the huge amount of data involved, a basic requirement is to minimize the human interference in the DCRM process. The progress in document imaging hardware and software has automated a large part of the process.

- High-speed scanners with Automatic Document Feeders (ADFs) have automated the raw data acquisition step.

- Automatic document layout analysis [3][4] and image compression techniques (for example, JBIG2 and JPEG2000, as well as more mature G4 and JPEG) have enabled the automatic creation of compact and hi-fidelity electronic documents.

- Advancements in Optical Character Recognition (OCR) and classifier combination [5][6] make it possible to regenerate the text information for indexing and searching with sufficient accuracy. Combined with the common “text-behind-image” electronic book formats, the manual correction of OCR results can be largely eliminated.

However, there is still one bottleneck to be conquered — the automatic construction of navigation information for long electronic documents. It is quite common that a book or journal has hundreds of pages. For paper-based documents, people are used to looking up the Table of Contents (TOC) and manually flipping to the correct pages. For electronically originated documents (for example, Word files or PDF files “distilled” from Word files), the built-in cross-reference information can help the reader to navigate through the contents. When the reader clicks on the text in the TOC pages, he/she will be led to corresponding individual starting pages of chapters or articles. In contrast, the raw re-mastered electronic book is simply a big file containing hundreds of pages, which can only be read one by one. Even with the latest devices like the Tablet PC, the reader’s experience will not be pleasant if he/she has to read sequentially, and awkwardly go back and forth to locate the content of interest. Thus, it is desirable to automatically build navigation capabilities into re-mastered digital books.

In the broad sense, this problem falls into the scope of document logic structure analysis. However, there are several unique desired characteristics when dealing with re-mastered books, which are not emphasized in previous research:

- The need for generic and template-free solutions.

A popular previous approach is to construct a structure template for a specific category of documents offline, and then to match the input document against the template to determine the logical structure [9][10]. This approach is not practical when we are facing a diverse and unknown variety of materials. Few books share the same template and it makes little sense to manually build a template that can only be used on a couple of books.

- The need for robustness against OCR errors and limitations.

Some existing methods utilize concrete text formatting information. For example, the article titles should be "30-point bold Times Roman." However, state-of-the-art OCR software cannot accurately extract text format information, such as font, font size, and spacing. Other methods [7] heavily depend on the page numbers printed on the TOC pages. If one page number is incorrectly recognized, the whole link will be lost.

- The need for proper granularity in long documents.

Usually electronic books are long documents, easily over several hundred pages in length. For such documents, we are mostly interested in the high-level structures, such as the chapters or articles, because the reader of an electronic book usually wants to find out the interesting chapters or articles and reading through them. In contrast, substantial existing research concentrates on finding out the very detailed structure of individual articles or even pages, such as where the abstract is and where a figure is referenced [10]. Consequently, these methods cannot provide the needed high-level navigation capabilities.

Section 2 introduces the core navigation construction algorithm based on text matching. Section 3 describes the implemented system and experimental results. Section 4 gives a summary and points to other related work and possible future directions.

2. TOC-BASED NAVIGATION CONSTRUCTION ALGORITHM

2.1. Basic idea

In this paper, we introduce a TOC-based algorithm to automatically add navigation capabilities to re-mastered books. Compared with existing logical structure analysis using TOC information [8], the proposed method does not analyze the TOC pages separately. Instead, the TOC pages are matched against the non-TOC pages throughout the whole process. Consequently, the proposed algorithm enjoys the above three desired characteristics. The basic idea relies on two criteria that are true for most long electronic books. First, there is an existing TOC in the document. The TOC need not be explicitly labeled, and one of our goals is to locate the TOC pages. Second, the text (for example, the article names and author names) in the TOC also appears on the start pages of individual chapters/articles.

Thus, the following steps are required:

- Identify the TOC pages.

- Segment the TOC pages into text regions.

- Map each text region with a chapter or an article.

In reality, the above subtasks are intertwined: The TOC pages are pages that have many regions linked with body pages, and one critical criterion of segmenting a TOC page into regions is that the text in one region appears on the start page of one chapter or article. Based on this observation, the proposed algorithm takes advantage of the result from one subtask to confirm the other subtasks. This tight coupling poses a significant difference between this work and earlier work that operates without feedback between steps [10].

Although this idea sounds straightforward, it is challenging to implement a robust and efficient algorithm. First, there can be many false matches between two pages, since we do not know in advance where the article titles and the author names are. Second, OCR errors may bring down otherwise perfect matches.

2.2. Algorithm design

This section describes the proposed solution. On the top level, we first select a set of pages as candidate TOC pages. For example, the first twenty pages are chosen as candidates. There is a tradeoff between computation load and missing TOC pages. The more candidate TOC pages we choose, 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 employed an adaptive scheme. In the initial phase, we choose the first twenty pages as TOC candidates. If the last few pages are confirmed to be the real TOC pages, we will expand the candidate set by another twenty pages.

Then each candidate TOC page will be matched against all of the pages following it. If the confidence score for that candidate page is above a certain threshold, that page will be confirmed. Simultaneously, the links from text regions on that page to title pages will be constructed. In order to efficiently match a candidate TOC page with the other pages, the following techniques are used:

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

Once such a dictionary (see Figure 1) is built, it takes much less time to find out if a word is on the candidate TOC page because the search is a nonlinear tree search instead of a sequential linear search. The experiment has shown a three-fold speedup using this technique.

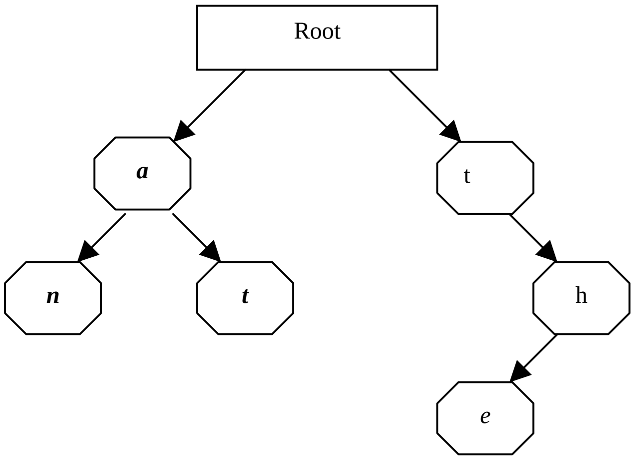


Figure 1: Tree dictionary structure based on TOC Pages

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

- Represent the candidate TOC page as a directional graph.

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 2 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 2 Range (4,6) corresponds to the phrase “person who studies”.

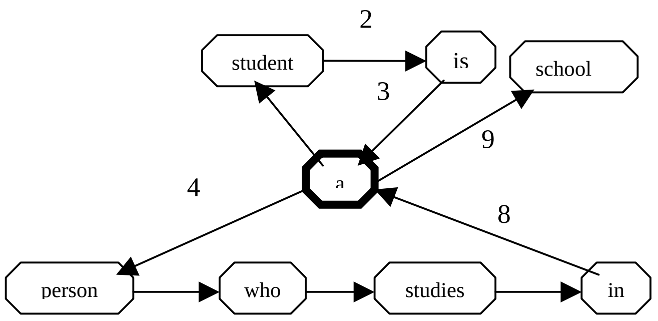


Figure 2: Graph description of TOC pages

- Divide the candidate TOC page into text chunks.

We are counting on the initial assumption of the replication of TOC phrases and chapter start page phrases. Therefore, a meaningful text chunk (called “Range” in this algorithm) in the TOC page should have a good match in the body pages. The following algorithm is used to detect the Ranges:

```
RangeList:= {}

foreach <Word>∈ BodyPages

    Find the longest match starting with Word between BodyPages and TOC. We put this match into a
    Range (s,e).

    If (s,e) overlaps with an existing Range in RangeList, the two Ranges are merged. If (s,e) does not
    overlap with any existing element in RangeList, we insert it into the RangeList.

endfor
```

- Evaluate the match results and finalize the links. The raw matching scores are numerical values. In the case of occasional OCR errors, the scores can be lower than those without OCR errors, but still high enough to find the links.

As an initial result from the above steps, it is completely possible that one Range will be linked to multiple body pages. Of course, we should keep only one link for each Range. So a “winner-take-all” strategy is adopted in the evaluation stage. All of the links to a Range are scored based on several factors, including the match length and the ordering of body pages. The best link is kept and the other links are deleted. Then the sum of the link scores on a candidate TOC page will be used to decide if it is a real TOC page. If a candidate TOC page is confirmed as a real TOC page, all of the links will be kept in the navigation system.

3. NAVIGATION SYSTEM OF ELECTRONIC BOOKS

Once the TOC pages are detected and the linking is identified, navigation capabilities can be automatically embedded into re-mastered books. Our experimental system creates PDF files. But it can also be easily amenable to other formats such as DjVu.

The navigation is added to the PDF files in two ways. First, a bookmark is automatically generated for each chapter or article (left panel of Figure 3). The text description of a bookmark comes from the Range(s) pointing to the chapter or article. When the user clicks on a bookmark, the window will display the related chapter or article. The detected TOC pages are also listed as bookmarks to facilitate navigation. Second, the Ranges on the TOC pages are marked out and linked to individual chapters or articles (see the right panel of Figure 3). When the user clicks on a text region on a TOC page, the other panel will display the linked body pages.

This system has been tested on re-mastered journals and books from MIT Press, some of which are available from [2]. It is difficult to come up with a very accurate measurement of the system's performance. For example, only part of the article name and/or the author name is linked to a title page because of OCR errors or even the documents themselves. In Figure 3, we can notice that the word "OF" in the title of Chapter 14 is not included in the link. Interestingly enough, this is because the title on the start page of that chapter actually uses "ON" instead of "OF", which should be considered as a typo or inconsistency in the original printed book. Consequently, "OF" is not part of the matched Range. Obviously, minor problems like this do not affect the user's experience. However, when the start pages are incorrectly inserted or deleted, they will negatively affect the user's navigation experience. Table 1 shows the testing results on 30 books and 7 journals published by MIT Press. The third column lists the error rate of detected start pages, including deletion and insertion errors. It can be seen that the error rate on journals is much lower than that on books. The major reason is that there is more information for matching TOC and start pages in journals, due to usually multiple author names and long article names. In the case of books, the available information is limited to chapter names, which can be as short as "Chapter 5".

In order to make the system more robust against OCR errors and other mismatches, we have also introduced the following heuristics into the algorithm. The weighting of these factors are experimentally tuned.

- The phrases in larger fonts are more likely to be titles. As mentioned earlier, OCR systems are not very accurate at detecting the absolute formatting information. However, font size estimation based on the bounding boxes of the text can still provide us with relative formatting information, which can be used to improve the analysis.

- Title phrases usually occupy the whole lines; namely, include the first and the last word of a line.

- Matching a long word is more significant than matching a short word.

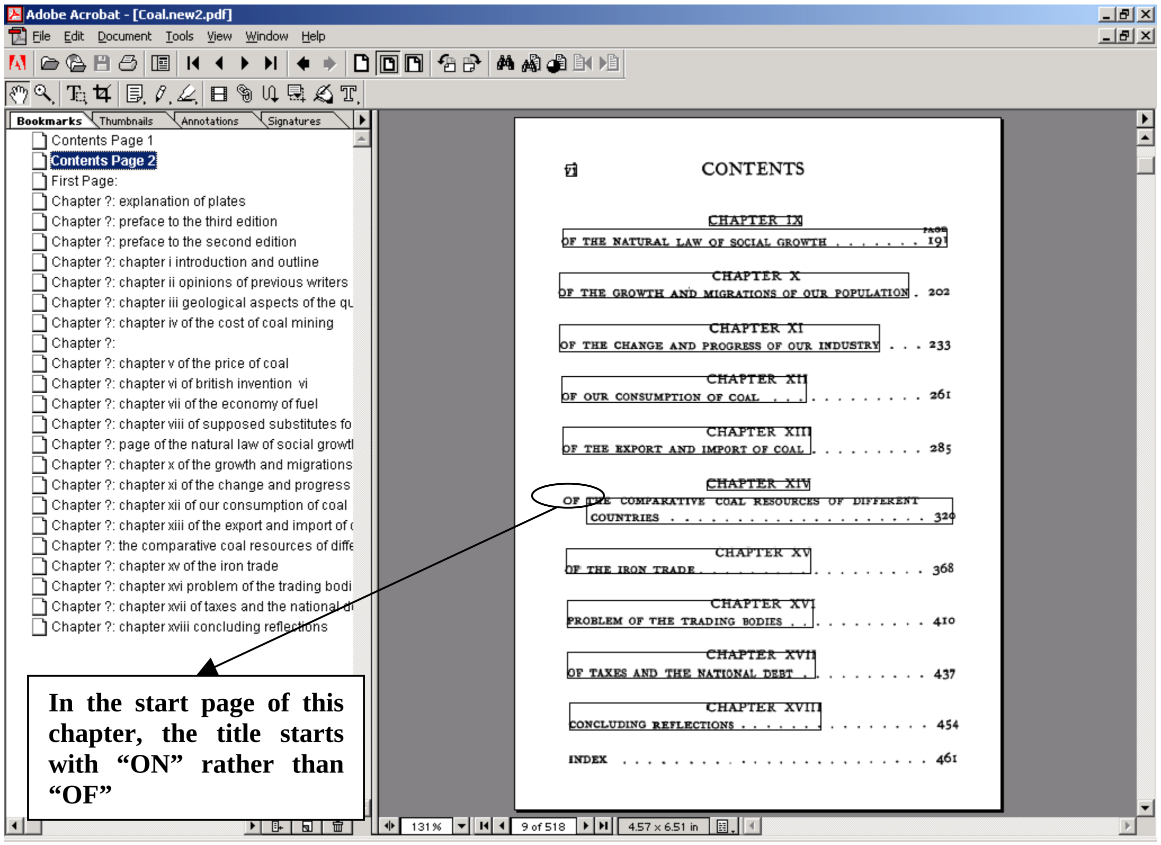
- Matching page numbers printed on TOC pages with those on the title pages.

We have also investigated how the quality of the OCR engine affects the linking algorithm. Figure 5 displays the results based on two OCR engines (see 5] for more details on the engines). Engine A has a character error rate of 1.1% while Engine B has an error rate of 0.46%. Obviously, more words on the TOC page are covered in the detected Ranges with the use of Engine B. In addition, one article is completely left out when Engine A is used. In conclusion, a superior OCR engine can still benefit the proposed linking algorithm, although the proposed algorithm is already less sensitive to OCR errors compared with many other methods. One possible future research topic is to quantitatively study how the linking error rate will be affected by OCR error rate.

As for the computation time, the 518-page book shown in Figure 3 takes 118 seconds to get the linking information and the 168-page journal shown in Figure 5 takes 49 seconds. The computer used is a 733 MHz

Pentium III desktop running Windows 2000. The time spent on linking is only a fraction of the time spent on the actual re-mastering process of converting TIFF files to PDF files. For example, the re-mastering (image processing only without the actual OCR) of the 518-page book takes about 15 minutes, and the re-mastering of the journal takes about 5 minutes.

Once the linking information is identified, the Adobe PDF Library [12] is used to write the bookmarks and links into the PDF file.

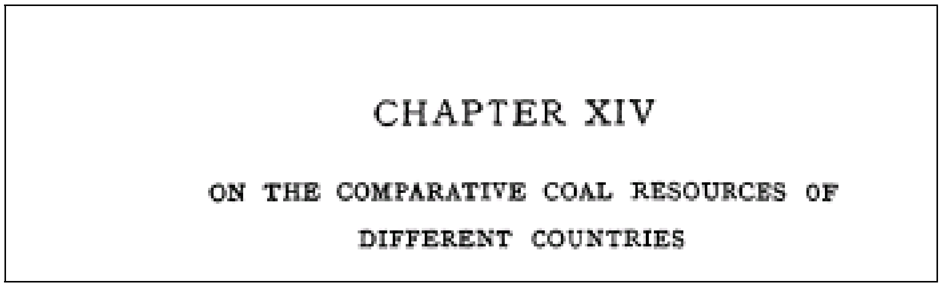

Adobe Acrobat - [Coal.new2.pdf]
File Edit Document Tools View Window Help
Bookmarks Thumbnails Annotations Signatures
131%
9 of 518
4.57 x 6.51 in

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 ?:
Chapter ?: chapter v of the price of coal
Chapter ?: chapter vi of british invention vi
Chapter ?: chapter vii of the economy of fuel
Chapter ?: chapter viii of supposed substitutes fo
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 c
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 di
Chapter ?: chapter xviii concluding reflections

In the start page of this chapter, the title starts with "ON" rather than "OF"

vi CONTENTS

Figure 3: Two ways of navigation in PDF files



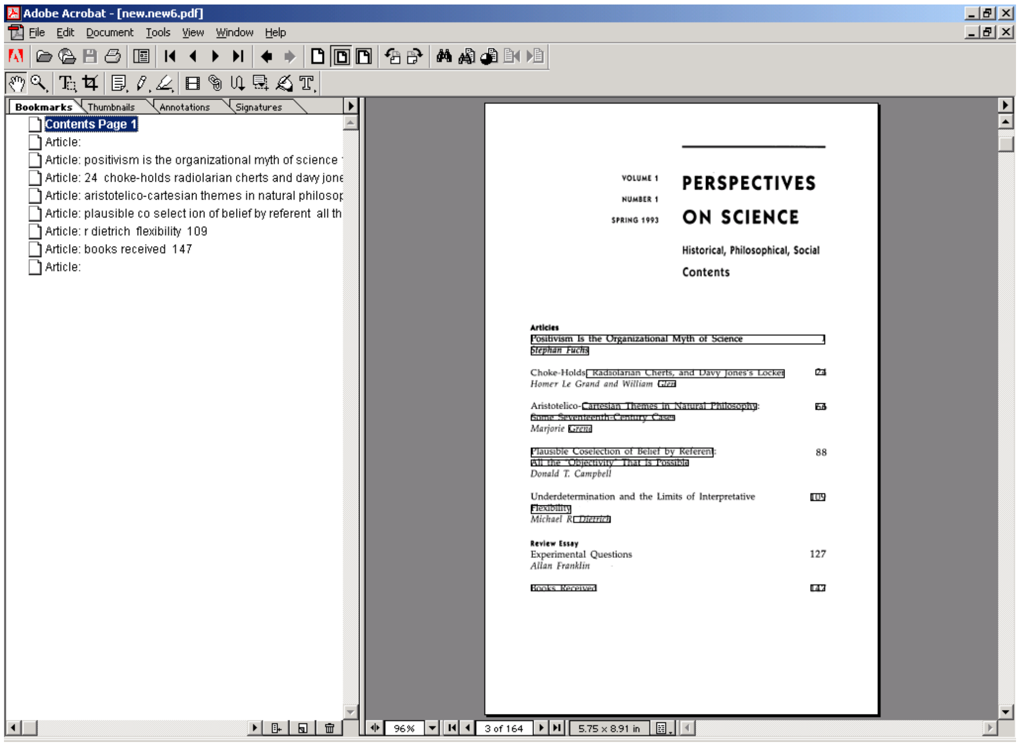
CHAPTER XIV

ON THE COMPARATIVE COAL RESOURCES OF DIFFERENT COUNTRIES

Figure 4: Start page of Chapter 14

Table 1: Testing results

Type	Number	Error Rate (Insertion plus deletion)
Academic Book	30	8.2 %
Academic Journal	7	2.0%


Adobe Acrobat - [new.new6.pdf]
File Edit Document Tools View Window Help
Bookmarks Thumbnails Annotations Signatures
96%
3 of 164
5.75 x 8.91 in

Contents Page 1
Article:
Article: positivism is the organizational myth of science
Article: 24 choke-holds radiolarian cherts and davy jone
Article: aristotelico-cartesian themes in natural philosop
Article: plausible co select ion of belief by referent all th
Article: r dietrich flexibility 109
Article: books received 147
Article:

VOLUME 1 NUMBER 1 SPRING 1993

PERSPECTIVES ON SCIENCE

Historical, Philosophical, Social

Contents

Articles

Positivism Is the Organizational Myth of Science 1
Stephan Fuchs

Choke-Holds, Radiolarian Cherts, and Davy Jones's Locker 24
Homer Le Grand and William Glen

Aristotelico-Cartesian Themes in Natural Philosophy: 66
Some Seventeenth-Century Cases
Marjorie Grene

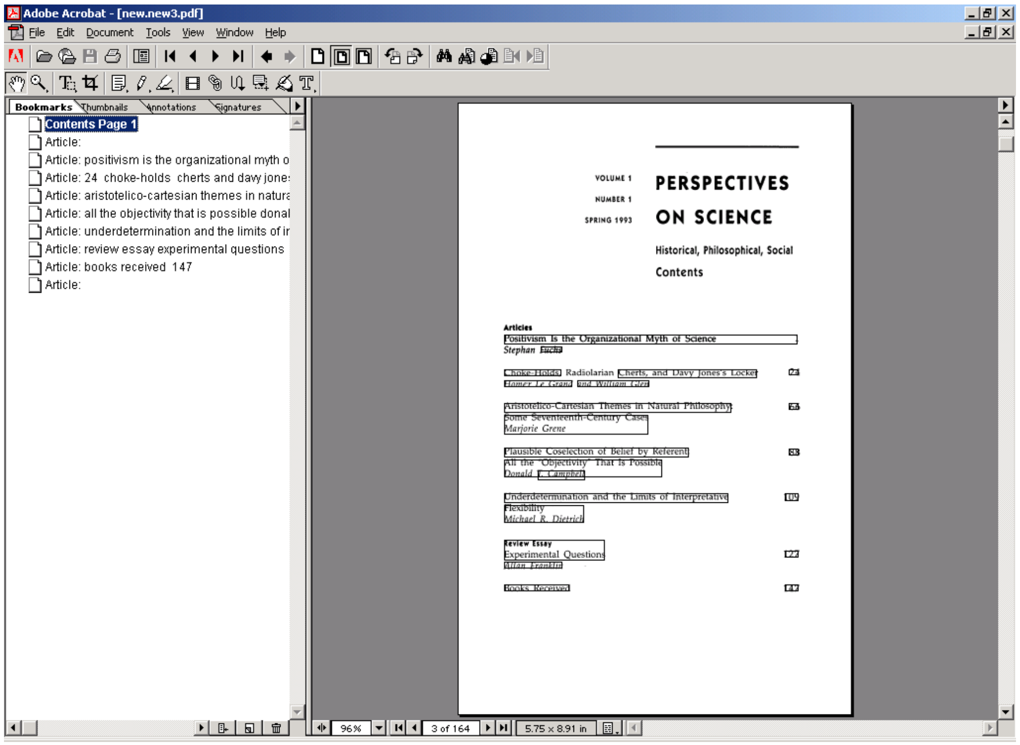
Plausible Coselection of Belief by Referent: 88
All the "Objectivity" That Is Possible
Donald T. Campbell

Underdetermination and the Limits of Interpretative 109
Flexibility
Michael R. Dietrich

Review Essay
Experimental Questions 127
Allan Franklin

Books Received 147

(a) Linking Result Based on OCR Engine A. Here we can see that the chapter on Page 127 has no link at all due to OCR errors.


Adobe Acrobat - [new.new3.pdf]
File Edit Document Tools View Window Help
Bookmarks Thumbnails Annotations Signatures
96%
3 of 164
5.75 x 8.91 in

Contents Page 1
Article:
Article: positivism is the organizational myth o
Article: 24 choke-holds cherts and davy jone
Article: aristotelico-cartesian themes in natura
Article: all the objectivity that is possible donal
Article: underdetermination and the limits of ir
Article: review essay experimental questions
Article: books received 147
Article:

VOLUME 1 NUMBER 1 SPRING 1993

PERSPECTIVES ON SCIENCE

Historical, Philosophical, Social

Contents

Articles

Positivism Is the Organizational Myth of Science
Stephan Fuchs

Choke-Holds, Radiolarian Cherts, and Davy Jones's Locker 24
Homer Le Grand and William Glen

Aristotelico-Cartesian Themes in Natural Philosophy: 66
Some Seventeenth-Century Cases
Marjorie Grene

Plausible Coselection of Belief by Referent: 88
All the "Objectivity" That Is Possible
Donald T. Campbell

Underdetermination and the Limits of Interpretative 109
Flexibility
Michael R. Dietrich

Review Essay
Experimental Questions 127
Allan Franklin

Books Received 147

(b) Linking result based on OCR Engine B

Figure 5: Impact of OCR on linking

4. CONCLUSIONS

In this paper, we have analyzed the need for a generic and robust system to automatically construct navigation links into re-mastered or unlinked electronic books. We have also applied the core text-matching algorithm to other document structure analysis tasks, such as journal splitting [11]. In addition, the 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 frequently pointed to by other pages. In this paper, the TOC pages are referenced by other body pages. However, there is a fundamental difference between this work and Web link mining. The linking information is explicitly marked with special HTML tags in Web link mining, while here the linkage is only implied in the form of repeated sentences. Extending the linking technology described here to identifying "high interest" pages is a possible future research direction. Another possible future research direction is to use deeper natural language processing capabilities. For example, we can extract key phrases on the body pages and locate those key phrases on the TOC pages.

ACKNOWLEDGEMENT

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

REFERENCES

1. L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. L. Cun, "High Quality Document Image Compression with DjVu," *Journal of Electronic Imaging*, **7(3)**, pp. 410-425, 1998.
2. MIT Press Classics Press Release, http://mitpress.mit.edu/main/feature/classics/MITPClassics_release.pdf.
3. F.M. Wahl, K.Y. Wong, and R.G. Casey, "Block Segmentation and Text Extraction in Mixed/Image Documents," *Computer Vision Graphics and Image Processing*, **Vol 2**, pp.375-390, 1982.
4. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis Machine Intelligence*, **22(8)**, pp. 888-905, 2000.
5. 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, Jan 2002.
6. A.F.R. Rahman, M.C. Fairhurst, "Multiple Classifier Decision Combination Strategies for Character Recognition: A Review", *International Journal of Document Analysis and Recognition*, **5(4)**, pp.166-194, 2003.
7. A. Belaïd, "Recognition of Table of Contents for Electronic Library Consulting," *International Journal on Document Analysis and Recognition*, **4(1)**, pp. 35-45, 2001.
8. 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*, Ulm, Germany, pp. 1048-1054, Aug 1997.
9. G. Nagy, S. Seth, and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals," *Computer*, **25(7)**, pp. 10-22, 1992.
10. 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, Aug 1995.
11. X. Lin, "Text-mining Based Journal Splitting," *Proc. of 7th International Conference on Document Analysis and Recognition*, pp. 1075-1079, Edinburgh, UK, Aug 2003.
12. Adobe Systems Inc., "Acrobat Core API On-line Reference for Acrobat Exchange 3.0," 1998.