Table of Contents Recognition for Converting PDF Documents in E-book Formats

Simone Marinai
DSI - Università di Firenze
Firenze, Italy
simone.marinai@unifi.it

Emanuele Marino
DSI - Università di Firenze
Firenze, Italy
marino@dsi.unifi.it

Giovanni Soda
DSI - Università di Firenze
Firenze, Italy
giovanni.soda@unifi.it

ABSTRACT

We describe one tool for Table of Content (ToC) identification and recognition from PDF books. This task is part of ongoing research on the development of tools for the semi-automatic conversion of PDF documents in the Epub format that can be read on several E-book devices. Among various sub-tasks, the ToC extraction and recognition is particularly useful for an easy navigation of book contents.
The proposed tool first identifies the ToC pages. The bounding boxes of ToC titles in the book body are subsequently found in order to add suitable links in the Epub ToC. The proposed approach is tolerant to discrepancies between the ToC text and the corresponding titles. We evaluated the tool on several open access books edited by University Presses that are partner of the OAPEN EcontentPlus project.

Categories and Subject Descriptors

I.7.4 [**Document and Text Processing**]: Electronic Publishing; H.3.1 [**Content Analysis and Indexing**]: Indexing Methods
Keywords PDF, E-book conversion, Table of Content

1. INTRODUCTION

Nowadays, larger publishing houses adopt document engineering best practices in the internal workflows. Publications are handled with markup languages such as LaTeX or XML from the beginning of the processing flow. In this way it is easier to re-format the documents towards re-flowable formats such as XHTML or Epub.

On the opposite, smaller publishing houses, for instance in the area of University Presses, sometimes attach the electronic publishing of monographs with a customized policy gathering the manuscript from authors in standard word processing formats and then directly generating the camera ready with state of the art publishing tools. The main objective in this case is to produce high quality editions, but there is little interest to perform an accurate tagging during the book production.

DocEng2010, September 21–24, 2010, Manchester, United Kingdom.

In the last few years, the need for alternative formats with respect to the PDF emerged as an interesting feature of many E-book readers. Even if some E-book devices, such as the Amazon Kindle and the Apple Ipad, allow users to display the PDF files, the reading experience that can be achieved with re-flowable and re-sizable formats is significantly better. Portable Document Format (PDF) is designed to allow users to exchange, view, and print electronic documents preserving their look in the most common architectures. Several converters are available either open-source (e.g. the Calibre system) or commercial (such as the Mobipocket Creator). A good survey of related techniques for reverse engineering of PDF documents can be found in [2]. A flexible method for detecting and understanding tables in PDF files is discussed in [3]. In [5] we described one system for the automatic extraction of administrative metadata from PDF documents for documents in digital libraries. Other techniques have been developed to extract the ToC especially from scanned documents [1] [4].

In this paper, we focus on the semi-automatic extraction of the Table of Contents (ToC) from PDF books. This is a sub-task of our current effort towards the development of tools aimed at converting digital born PDF books in a format readable by E-book devices. Among various alternatives, the Epub format is nowadays believed to emerge as a widely used one. To build the ToC most systems for PDF to E-book conversion rely on regular expressions or XPath templates to search suitable XHTML structures in the document body. However, this technique can be applied only when tags that unambiguously identify ToC entries can be found in the document.

The paper is organized as follows. In Section 2 we discuss the problems related to the conversion of PDF documents in E-book formats. In Section 3 we analyze the proposed tool that has been experimentally tested as discussed in Section 4. Final conclusions are drawn in Section 5.

2. CONVERSION OF PDF BOOKS TO EPUB

In recent years some tools have been developed for the conversion of digital works (e.g. in PDF format) in formats readable by E-book devices. The ToC extraction is an important task of this conversion since it allows an easy navigation between its parts (e.g. chapters). Basically, the Epub format is based on HTML. An Epub file is one compressed archive that contains some files with document metadata as well as HTML files (for instance one for each chapter) with images and CSS stylesheets when needed. Moreover, in some cases there is a special file that contains the document ToC,

with explicit pointers to the related book parts. In E-book readers it is very important to have an electronic ToC since the various components of the book are moved on different pages when the text-size is modified. Since the pages are resized when modifying the font size, the ToC must point to the paragraph related to the chapter title and not simply to the page containing this title (as performed by several ToC extraction algorithms [1]). This requirement is considered by the method described in this paper.

In our system the PDF file is analyzed with the JPedal library that allows us to extract the textual content of the document, with additional information such as the text location, font, and size. Unfortunately, the font information is seldom useful to identify the ToC titles and therefore we did not use this information in our tool. After identifying the ToC structure (Section 3) the book conversion is performed by first extracting the paragraphs in each page and then identifying the HTML paragraph that corresponds to each title. At the end, for each paragraph containing one ToC title we add a link from the related ToC entry.

Besides ToC extraction, there are other tasks that are essential for an effective PDF conversion, especially when dealing with scientific and technical documents. Among other kinds of objects to be extracted we are currently working on the identification of tables and equations that are difficult to handle by state of the art PDF converters. This research is not addressed in the present paper.

3. TOC IDENTIFICATION AND ANALYSIS

In our system, the ToC extraction is a semi-automatic since a human operator is needed to inspect and modify the extracted ToC in case of errors. The proposed system is based on six main steps:

1. Identification of the ToC pages.
2. Manual markup of ToC pages in case of missing pages.
3. Identification of *potential titles* in the ToC.
4. Search for *potential titles* in the book and identification of the *target titles*.
5. User validation or editing of the automatic identification.
6. Export in Epub format.

In this paper, we describe only the automatic recognition (steps 1, 3, and 4). One Potential Title is one string in the ToC that could correspond to one actual title. For instance, in Figure 1 the entry pointing to page 60 has two potential titles: “forma di reiterazione” and “1.2 descrizione. Posizioni diverse nel TV-Duell come forma di reiterazione”. The second is the *actual title*. One *potential title* for each entry (page number) is defined as actual title when its text is found in the book.

From a preliminary analysis of the books to be processed it turned out that an exact search is not suitable to look for *potential title* in the book this problem is due to differences both in the style and in the printed text between the ToC entry and the corresponding *target title* in the book body. For instance, the chapter author could be at the beginning or at the end of one of the two titles. Moreover, to extend the use of the system to scanned books it seems appropriate to allow spelling differences.

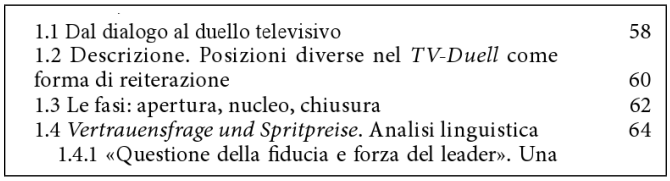

Figure 1: Excerpt from a ToC page.

3.1 ToC pages identification

The ToC is identified by looking for pages containing a sequence of numbers aligned on the right part of the page, with increasing values. The search for such pages is made on the first (N_{first}) and last (N_{last}) pages of the book. This heuristic search is suitable for digital-born books, but can be adapted to deal with scanned documents as well
In the following, we refer to the page numbers printed in the ToC as *link-page (lp)* and to the actual page position in the file as *target-page (tp)*. The two values are in general different since the file frequently starts with the cover and contains some non-numbered initial pages. One simple approach to identify the *tp* corresponding to each *lp* could be to look for the *lp* numbers in the book body and then assign this page to *tp*. However, this approach will not work when the page number is missing, the sequence of page numbers in the book contains empty pages, or one single file corresponds to multiple volumes. For these reasons a more complex solution has been developed as described in the following.

3.2 Identification of Potential Titles

Once the ToC pages are located, the text corresponding to each *lp* must be identified. Apart from simple cases, when one and only one text line corresponds to one *lp*, for each *lp* we extract all the *potential titles* that could be associated to it. The *potential titles* are incrementally built starting from the text line aligned with the current link-page. Let lp_i be the i-th link-page belonging to the j-th text-line in the ToC page and let Str_j be the text without the page number. The *potential titles* (referred to as $Entry_{i,j}$) corresponding to lp_i are build looking backward until we reach the line corresponding to lp_{i-1}:

$Entry_{i,0} = Str_j$;
$Entry_{i,1} = Str_{j-1} + Entry_{i,0}$;
...
$Entry_{i,k} = Str_{j-k} + Entry_{i,k-1}$

In the example shown in Figure 1, there are two possible Entries for the $lp = 60$:

$Entry_{16,0}$ = ”forma di reiterazione”;
$Entry_{16,1}$ = ”1.2 descrizione. Posizioni diverse nel TV-Duell come forma di reiterazione”;

Each *potential title* is then searched in the volume. The best matching $Entry_{i,j}$ for lp_i is used to identify the tp_i.

3.3 Tri-gram indexing

In many cases, the search for the *potential titles* in the book body can not be based on exact string matching techniques. Significant differences can occur between the *potential titles* and the *target title* printed in the book. We

adopted one technique that allows us to implement a inexact string matching by using trigram indexing of the whole book text.
Trigrams are sequences of three consecutive characters extracted from each word (for instance the word 'three' generates the following trigrams: 'thr', 'hre', and 'ree'). *Target titles* could be searched by associating each trigram with its text-line and then looking for the book text-line containing the maximum number of trigrams in the *potential title*. In this way it is possible to deal with spelling variations, but titles spanning more lines in a different way with respect to the ToC are difficult to handle.

We therefore associate each trigram with the corresponding words rather than with the text-lines. The *target titles* are subsequently obtained by grouping together the words belonging to one title even if they belong to different text-lines. In the inverted index for each occurrence of a trigram we store the bounding box, the page number, and the word string that contains the tri-gram. Before extracting the trigrams we remove from the indexed terms punctuation marks and other characters such as brackets.

3.4 Search for Target Titles

Target titles are searched in the book according to the printed ToC. We start by searching the title corresponding to lp_0 (we look for all the corresponding $Entry_{0,j}$) and we then proceed with the other lp_k.
If tp_k is the page assigned to lp_k, we search tp_{k+1} by searching $Entry_{k+1,j}$ starting from page tp_k (since two titles could be in the same page). If no page corresponding to lp_k is found one hint on the title location will be provided later with an interpolation with pages identified.

For each lp_i we identify one and only one $Entry_{k,j}$ ($j = 0, \ldots, m-1$) that corresponds to lp_k. This is the *actual title* ($Title_k$). When we identify the $Title_k$ we can link it with the *target title* adding a *bookmark* in the PDF file and including this information in the Epub ToC.

The search for $Title_k$ is made in two steps. First, we assign to each page one *score* $\in [0, 1]$. Second, we check the pages with highest scores, and we find the *target title* position.

3.4.1 Page selection

We first search all the occurrences of the trigrams obtained from $Entry_{i,j}$. Let W_h ($h = 1, .., N$; N number of book pages) be the set of words in the page p_h with at least one trigram of $Entry_{i,j}$. We assign one score to each page p_h on the basis of a similarity measure of each word of $Entry_{i,j}$ with all the words in the list W_h. Let $Word_1$ and $Word_2$ be two words to be compared; let $TR(Word_i)$ be the set of trigrams that can be generated by the word $Word_i$. The similarity between the two words is computed as:

$$S(Word_1, Word_2) = \frac{\#(TR(Word_1) \bigcap TR(Word_2))}{Max(\#(TR(Word_1)), \#(TR(Word_2)))}$$

We can then assign a score to each word in $Entry_{i,j}$, by looking for the closest word in the page. Let $WordE_y$ be one word of $Entry_{i,j}$ and $Word_x$ be one word in W_h the score of $WordE_y$ is:

$$Sc(WordE_y) = \max S(WordE_y, Word_x) \quad (1)$$

The score of page p_h is computed by

$$Score(h) = \frac{\sum_{y=0}^{s-1} Sc(WordE_y)}{s} \quad (2)$$

where s is the number of words in $Entry_{i,j}$.

3.4.2 Building the Title Bounding Box

The Title Bounding Box (TBB) are the coordinates of the upright rectangle enclosing the *target title*. We build TBB by joining the Word Bounding Boxes (WBB) that are stored in the list W_h obtained from the search of a given $Entry_{i,j}$. We first sort the WBB in page p_h on the basis of the reading order: left to right and top to bottom. According to the relative position of the WBB in the page we can have four possibilities:

a) There is no text on the left of WBB, but there is text on the right.

b) There is text on the left of WBB, but there is no text on the right.

c) There is text both on the left and on the right of WBB.

d) There is no text both on the left and on the right of WBB.

One WBB marked as 'a' can contain the first word of one title, similarly one WBB marked as 'b' can contain the last word of one title. With a formalism based on regular expressions, we can represent a title as:

$$Title = \begin{cases} d & \text{single word} \\ ac^*b & \text{single text line} \\ (d^*|(ac^*b))^* & \text{more text line} \end{cases} \quad (3)$$

We therefore scan the list W_h and look for all the subsequences having the previous structure. To reduce the number of false positives, the sequences of WBB are accepted only if the WBB to be joint are close in horizontal and vertical directions. When we find one subsequence, we build the TBB by computing the union of the corresponding WBB. This procedure returns one lists of titles belonging to page p_h.

3.4.3 Target title identification in the page

In this step we analyze the Title Bounding Box and we build the *target title*. We analyze the TBB of all the possible titles in page p_h (Section 3.4.2) and we decide that one TBB can contain the *target title* if the following conditions are satisfied:

1. The length of the string contained in TBB ($|TBB|$) is such that:
   $\frac{1}{2} \cdot |Entry_{i,j}| < |TBB| < 2 \cdot |Entry_{i,j}|$.
2. There is no text on the left and right sides of TBB.
3. The font size of the string in TBB is larger than the average font size in the book.

The TBBs satisfying the above properties can be selected as *target title* for a given link-page.

We therefore have a pair of one *potential title* $Entry_{i,j}$ and the corresponding $Target_{i,j}$. By checking the $Target_{i,j}$ we

can identify the most likely pair $Entry_{i,j}$ (in the ToC) and $Target_{i,j}$ (in the book). In some cases there are more than one *potential title* for one linkpage lp_i. The best *potential title* and the corresponding *target title* are identified according to the following algorithm:

If $(Page(Target_{i,j}) \neq Page(Target_{i,j+1}))$

then $Title_i = TMax(Score(Entry_{i,j}), Score(Entry_{i,j+1}))$

else If $(TBB(Target_{i,j}) = TBB(Target_{i,j+1})))$

then $Title_i = Entry_{i,j}$

else $Title_i = Entry_{i,j+1}$

where $TMax$ is a function that returns the Entry having the highest score. Note that if $Target_{i,j}.TBB = Target_{i,j+1}.TBB$ then $Entry_{i,j+1}$ (that is longer than $Entry_{i,j}$) contains some extra text that is not occurring in the *target title*. An example of this case can happen when $Entry_{i,j+1}$ includes a running head in the ToC page (such as "Contents" or similar).

3.5 Recovering missing titles

In some cases it is not possible to find the *target titles* for some link-pages. This is due for instance to entries in the ToC that are very different from the corresponding *target titles*.

The system marks the title as *not found* and provides, after processing the whole ToC, some tentative hypothesis about the corresponding page. However, in this case it is not possible to build the TBB and identify the *target title*. There are two approaches to guess one possible matching page: selecting the page with the highest score or computing an interpolation based on pages recognized with enough confidence.

In our system we developed an interpolation-based technique that sets a missing tp on the basis of neighboring *target titles* found. Let us suppose that no title has been identified for lp_i, but we have found tp_{i-1} corresponding to lp_{i-1}. Without additional information, we can suppose that a fixed offset allows to go from $lp(i-1)$ to $tp(i-1)$ and from $lp(i)$ to $tp(i)$. Therefore we can estimate tp_i as follows:

$$tp_i = lp_i + lp(i-1) - tp(i-1). \quad (4)$$

When the first tp is missing, we can estimate the offset from the next entries and apply this value to estimate $tp(0)$.

From the experiments summarized in the next section we can notice that the interpolation allowed us to recover many missing titles.

4. EXPERIMENTAL EVALUATION

In first experiments reported in this paper, we tested the system on a dataset containing 22 digital born books. 12 books came from the *Open Access* collection of the *Firenze University Press (FUP)* whereas the remaining 10 books have been downloaded from the Open Access collection of *Amsterdam University Press (AUP)*. In total, there are 1031 links to be identified. In Table 1 we summarize the results achieved pointing out the improvement that has been obtained with the interpolation step.

Dataset	Books	Links	Precision	Precision interpol.
FUP	12	412	92.96	97.09
AUP	10	619	90.79	97.90
Tot.	22	1031	92.55	97.5

Table 1: Precision on the two datasets without and with the interpolation.

5. CONCLUSIONS

In this paper we described a technique that we designed to extract the Table of Content from digital-born PDF books. The problem is at first quite simple, but some challenges should be considered in order to deal with differences between the printed ToC and the corresponding text in the book. This method is an important part of a more general tool that is aimed at converting PDF books in a format that can be read by E-book devices. Even if designed to work with digital-born books most techniques described in this paper can be used for scanned books as well.

Acknowledgements
This work is partially supported by OAPEN EContentPlus Project, co-funded under the EU 7th Framework Programme.

6. REFERENCES

[1] H. Déjean and J.-L. Meunier. On tables of contents and how to recognize them. *IJDAR*, 12(1):1–20, 2009.

[2] K. Hadjar, M. Rigamonti, D. Lalanne, and R. Ingold. Xed: a new tool for extracting hidden structures from electronic documents. In *DIAL '04. First Int'l Conference on Document Image Analysis for Libraries*, pages 212–224, 2004.

[3] T. Hassan and R. Baumgartner. Table recognition and understanding from PDF files. In *ICDAR 2007. Ninth Int'l Conf. on Document Analysis and Recognition*, pages 1143–1147, 2007.

[4] X. Lin and Y. Xiong. Detection and analysis of table of contents based on content association. *IJDAR*, 8(2-3):132–143, 2006.

[5] S. Marinai. Metadata extraction from pdf papers for digital library ingest. In *10th International Conference on Document Analysis and Recognition*, pages 251–255, 2009.