2011 International Conference on Document Analysis and Recognition

Conversion of PDF books in ePub format

Simone Marinai, Emanuele Marino, and Giovanni Soda

Dipartimento di Sistemi e Informatica
Università di Firenze, Italy
simone.marinai@unifi.it

***Abstract*—In the last years the interest in e-book readers is significantly growing. Two main document formats are supported by most devices: *PDF* and *ePub*. The *PDF* format is widely used to share documents allowing a cross-platform readability. However, it is not ideal for a comfortable reading on small screens. On the opposite, the *ePub* format is re-flowable and it is well suited for e-book readers.**

In this paper we describe a system for the conversion of *PDF* books to the *ePub* format aiming at inverting the text formatting made during the pagination. To this purpose, layout analysis techniques are performed to identify the book's table of contents and the main functional regions such as chapters, paragraphs, and notes.

***Keywords*-Ebook; ePub; Layout Analysis; PDF;**

I. Introduction

The *PDF* format is probably the most widely used to deliver documents that can be rendered consistently in a large spectrum of different devices. Although most e-book readers allow users to read and, in some cases, to annotate *PDF* documents, one important problem is the difference in size between the display and the most common paper formats. Most devices have a size of about 6 inches whereas the basic formats for printed documents are the $A4$ and the letter. Larger screen sizes (up to 10 inches) are available for tablets, but the higher cost and the reduced portability limit the usability of these devices for prolonged book reading.

On the opposite, re-flowable formats, such as the *XHTML*, allow the user to modify the font size while reading and subsequently adapt the information on the screen on the basis of the display area. The advantages of re-flowable formats are inherited by the *ePub* open standard that can be read by most e-book devices.

While the conversion of *PDF* documents containing only text, like novels, is relatively easy, the processing of works in other domains such as books in Human and Social Sciences is more difficult. In these works it is essential to identify the book structure (encoded in the Table of Contents, *ToC*) and the notes (either at the end of the page or at the end of the chapter/book). Another relevant problem is related to the suitable handling of bibliographic references. Besides *ToC* and notes 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 we are currently working on the identification of tables and equations that are difficult to handle by *PDF* converters. The latter research is not addressed in the present paper.

In this paper, we describe a system designed for the conversion of *PDF* books in the *ePub* format. In Section II we analyze the main differences of the two formats. The proposed system is described in Section III. The experiments are summarized in Section IV and some conclusions are reported in Section V.

II. Document formats

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 [1]. Some methods for detecting and understanding tables in *PDF* documents are discussed in [2] and [3]. In [4] we described a tool for the automatic extraction of administrative metadata from *PDF* documents in digital libraries. Other techniques have been developed to extract the *ToC* especially from scanned documents [5] [6], whereas a competition dedicated to Book Structure Extraction has been organized at ICDAR 2009 [7]. In [8] we described a method for the semi-automatic extraction of the Table of Contents (*ToC*) from *PDF* books that is the first sub-task of the system described in this paper.

The *ePub* format for e-books is widely used and supported by most devices. An *ePub* document is made by one *ZIP* archive that assembles some files that contain both document metadata and the main text content encoded in one or more *XHTML* files corresponding to separated chapters. Images and CSS stylesheets are included in other files. In most books the *ePub* file includes also an NCX file that describes the document's *ToC* and allows an easy navigation by pointing to the indexed parts of the book.

In the last few years, the need for alternative formats with respect to *PDF* emerged as an interesting feature of many e-book readers. Even if many e-book devices allow users to display the *PDF* files, the reading experience with re-flowable and re-sizable formats is significantly better.

In 2011 the new standard *ePub3* is planned to be adopted by the International Digital Publishing Forum [9] consortium that has been set up to define a standard for e-books. *ePub3* incorporates HTML5 and CSS3 and allows documents to include multimedia elements, such as video, and SVG objects.

III. Converting books

Nowadays, most publishing companies adopt document engineering tools to produce electronic books. Publications are handled with markup languages such as LaTeX or XML from the beginning of the processing chains. In so doing, it is possible to achieve an automatic conversion of the documents to new formats and in particular to re-flowable formats such as *XHTML* or *ePub*. On the other hand, books produced with the main purpose of a perfect rendering on the printed support, are only encoded in camera-ready *PDF* files.

The first step in the book conversion is the *PDF* parsing that is implemented (with the JPedal library) to extract the textual elements together with additional information such as the text position, font, and size. The conversion is then based on three main steps:

1) *ToC* identification and analysis
2) Identification of Notes and Illustrations.
3) Export in *ePub* format.

The automatic recognition of the *ToC* has been described in details in [8] and is summarized in Section III-A. In the subsequent Sections we describe the steps 2 and 3 and in particular the extraction of notes.

A. ToC identification and analysis

The assembly of an electronic *ToC* is particularly convenient in e-book readers because it allows to track the position of the book parts that are moved when the pagination is modified to comply with a user-requested font-size change. In an electronic book the *ToC* entries point to the actual chapter title and not only to the first page of the chapter.

The identification and processing of the Table of Contents (described in [8]) is based on the following steps:

- Identification of the pages containing the printed *ToC*.
- Identification of *potential titles* in the *ToC*.
- Search for *potential titles* in the book and identification of the *target* and *actual* titles.
- User validation of the *ToC*.

One *potential title* is one string in the printed *ToC* that could correspond to one title in the book's body (the *target title*). When this match is found, a potential title is labeled as an *actual title*. The *ToC* extraction is semi-automatic and a user is asked to check the extracted *ToC* and adjust it in case of errors. The user identifies, from the list detected by the system, what organization he/she would like to consider in the *ePub* document and therefore what parts of the book are to be split in separate files.

After identifying the *ToC* structure the book conversion is performed by first extracting the paragraphs in each page and then identifying the *XHTML* paragraph that corresponds to each title. At the end of the process a link is added from each *ToC* entry to the corresponding paragraph in the book. The reconstruction of paragraphs from *PDF* documents is complex in case of paragraphs that are split in two pages. This is more difficult when footnotes or other objects are intermixed in the main text. Additional problems are related to the conversion of mathematical equations, tables, and bibliographic entries but these items are not addressed in this work.

B. Processing of Notes and Illustrations

The second step of the system is performed with the following sub-steps:

1) Layout analysis: reconstructing text lines.
2) Layout analysis: paragraph identification.
3) Classification of paragraphs.
4) Processing notes.

The identification and extraction of illustrations is relatively simple in *PDF* documents. The layout preservation and the identification and processing of notes is more difficult and are described in the following.

1) Layout analysis: reconstructing text-lines: The first step in the layout analysis is the reconstruction of the textlines (*TL*) starting from the coordinates of each word. The words are first of all sorted on the basis of the reading order. We subsequently consider the *baseline* of the words to reconstruct the corresponding *TL*. After extracting the *TL*s the system performs the layout analysis on the basis of the horizontal projection profiles (only horizontal cuts are considered since we deal with single column books). During the *TL* building, we identify the illustrations, that are saved as *PNG* files and will be linked in the corresponding *XHTML* file.

We also compute the *page bounding box* that is used to identify the centered *TL*s (for instance belonging to title pages) and the indented *TL*s that probably mark the beginning of a new paragraph. Moreover, we compute some statistics on the type and size of fonts in the document and we deal with hyphenated words by merging subwords split with the '-' character at the end of one textline. This is required to generate a *reflowable* document.

2) Layout analysis: paragraph identification: At the end of the processing of the *TL* list we build the paragraphs that are exported in *XHTML* when producing the *ePub* file. While the *TL* construction is based on word aggregation in this step we perform a segmentation between contiguous *TL*s so as to generate a flux of paragraphs.

The central part of the *TL* segmentation is one procedure that given two *TL*s (tl_1 and tl_2) decides whether they belong to the same paragraph. In practice, we append *TL*s one after the other until two subsequent *TL*s have different features. The *TL* segmentation is designed considering features based on the font and on geometric information such as the average distance between *TL*s belonging to the same paragraph. This distance is computed dynamically on the basis of the height of the last *TL* assigned to the current paragraph. Another geometric feature is the indentation of a *TL* that is a criterion for splitting two paragraphs. Taking into account the font information, it is possible to separate two *TL*s having a different font, or having different alignments (for instance one *TL* centered and one not). Paragraphs that continue from one page to the next one are merged in a subsequent step.

3) Classification of paragraphs: At the end of the previous step the P list contains the extracted paragraphs with an associated class. This classification is required to produce one *ePub* document that is most similar to the original *PDF* not only for its content, but also for its appearance. The paragraphs in the P list are classified into one of the following classes:

- *Image*
- *ToC*
- *Title*
- *Chapter*
- *Running Head*
- *Footer*
- *Note*
- *Current Text*

The *Image* class is easy to identify, since images are stored as separate objects in the *PDF* file. The other classes correspond to textual paragraphs that should be rendered in different ways. The *ToC*, *Title*, and *Chapter* paragraphs are identified in the first steps that perform the *ToC* analysis. The *ToC* information is copied in the NCX file of the *ePub* archive. The *Title* class is assigned on the basis of geometric features. In particular, we scan the P list and look for the paragraph whose bounding box intersects one *actual title* found during the *ToC* extraction step. This search is limited to the page containing the *actual title*.

The class *Chapter* is required to insert the correct *break points* used to obtain one separated *XHTML* file for each chapter. There are two kinds of break points: those identified by the system and those defined by the user. The first are inserted between pages at the beginning of the book (before the *ToC*). In the remaining pages the break points in most cases correspond to chapters, but the user can define a different organization of the files in the *ePub*. To this purpose, the GUI presents the *ToC* entries identified to the user that chooses what entries should be considered as real break points.

The *Running Head* and *Footer* classes are identified on the basis of geometric and structural information. For instance, the *Running Head* class is assigned to the first paragraph of each page that is composed by a single *TL* and has different font with respect to the main text. The *Footer* class is identified in a similar way after removing the *Note* paragraphs.

The *Note* class is the most difficult to identify and to handle. Its processing is described in the next sections.

4) Processing notes: The notes to be located belong to three main categories:

- **A)** Notes printed at the end of the corresponding page ("footnotes").
- **B)** Notes printed at the end of each chapter.
- **C)** Notes printed at the end of the book, often in a separate chapter. The numbering of notes can be global or there can be a separated list for each chapter.

We consider two items related to notes: the reference to the note (*RefNote*) that can be found in the running text, and the referenced note (*Note*) that can belong to one of the previous three categories. The *RefNotes* are identified when the paragraphs are processed and the *XHTML* output is generated. In most cases, it is quite easy to find the *RefNote*. To this purpose we look for a number (currently we do not deal with other keys such as "*") with a font size smaller than the main text one: $fs_N/fs_T < \delta_F$, where fs_N is the number font size, fs_T is the main text font size, and δ_F is a threshold that is fixed to $\delta_F = 0.75$.

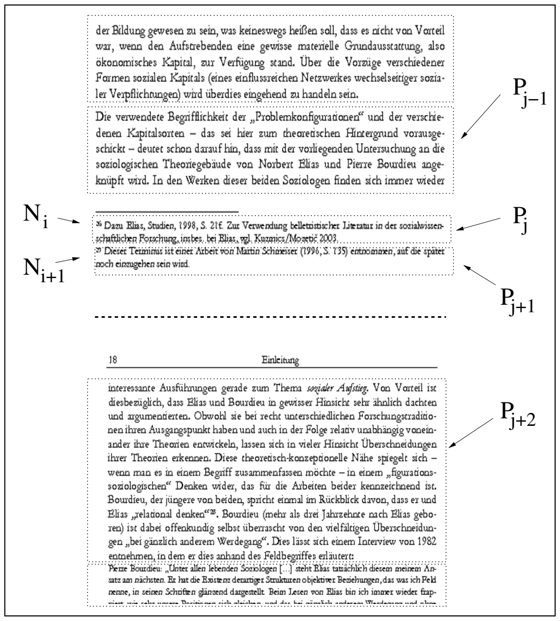
The identification of the notes is more complex. When the paragraphs are classified, we build the N list that contains all the potential notes. In addition to the pointer to the corresponding *current text* paragraph the list contains other information: the $Label$ (number of the note), the note font type, the $Label$ font type, the size of the note.

The potential notes are identified considering the paragraphs beginning with a number ($Label$). In so doing there can be several false positives, but this strategy is applied to maximize the recall. The false positives in the N list are due to the following main reasons:

- Numbered lists (including the Bibliography).
- Numbered sections not included in the printed *ToC* (therefore not labeled as *Title*).
- Numbered figure captions.

To locate the notes, the system scans the list of potential notes N and identifies the subsequences $N_1, ..., N_k$ corresponding to actual notes. In case of notes of type **B** or **C** the subsequence is made by consecutive paragraphs. On the opposite, footnotes (notes of type **A**) are intermixed with running text paragraphs. In this case we merge together the text paragraphs in two subsequent pages that are separated by k notes ($k \geq 1$). For notes at the end of the chapter or at the end of the book, the paragraph merging is not performed and the sub-sequences corresponding to each chpater are identified at once.

To identify actual notes, the system takes into account two main features. One is statistical and is based on a comparison of the paragraph font size with the $Label$ font size. The other


der Bildung gewesen zu sein, was keineswegs heißen soll, dass es nicht von Vorteil war, wenn den Aufstrebenden eine gewisse materielle Grundausstattung, also ökonomisches Kapital, zur Verfügung stand. Über die Vorzüge verschiedener Formen sozialen Kapitals (eines einflussreichen Netzwerkes wechselseitiger sozialer Verpflichtungen) wird überdies eingehend zu handeln sein.

Die verwendete Begrifflichkeit der „Problemkonfigurationen" und der verschiedenen Kapitalsorten – das sei hier zum theoretischen Hintergrund vorausgeschickt – deutet schon darauf hin, dass mit der vorliegenden Untersuchung an die soziologischen Theoriegebäude von Norbert Elias und Pierre Bourdieu angeknüpft wird. In den Werken dieser beiden Soziologen finden sich immer wieder

[26] Dazu Elias, Studien, 1998, S. 21f. Zur Verwendung belletristischer Literatur in der sozialwissenschaftlichen Forschung, insbes. bei Elias, vgl. Kuzmics/Mozetič 2003.

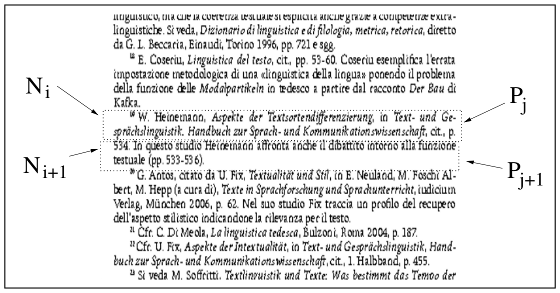
[27] Dieser Terminus ist einer Arbeit von Martin Schmeiser (1996, S. 135) entnommen, auf die später noch einzugehen sein wird.

18 Einleitung

interessante Ausführungen gerade zum Thema *sozialer Aufstieg*. Von Vorteil ist diesbezüglich, dass Elias und Bourdieu in gewisser Hinsicht sehr ähnlich dachten und argumentierten. Obwohl sie bei recht unterschiedlichen Forschungstraditionen ihren Ausgangspunkt haben und auch in der Folge relativ unabhängig voneinander ihre Theorien entwickeln, lassen sich in vieler Hinsicht Überschneidungen ihrer Theorien erkennen. Diese theoretisch-konzeptionelle Nähe spiegelt sich – wenn man es in einem Begriff zusammenfassen möchte – in einem „figurationssoziologischen" Denken wider, das für die Arbeiten beider kennzeichnend ist. Bourdieu, der jüngere von beiden, spricht einmal im Rückblick davon, dass er und Elias „relational denken"[28]. Bourdieu (mehr als drei Jahrzehnte nach Elias geboren) ist dabei offenkundig selbst überrascht von den vielfältigen Überschneidungen „bei gänzlich anderem Werdegang". Dies lässt sich einem Interview von 1982 entnehmen, in dem er dies anhand des Feldbegriffes erläutert:

Pierre Bourdieu: „Unter allen lebenden Soziologen […] steht Elias tatsächlich diesem meinem Ansatz am nächsten. Er hat die Existenz derartiger Strukturen objektiver Beziehungen, das was ich Feld nenne, in seinen Schriften glänzend dargestellt. Beim Lesen von Elias bin ich immer wieder frappiert, wie sehr unsere Positionen sich gleichen, und das bei gänzlich anderem Werdegang und ohne

Figure 1. P_{j-1} and P_{j+2} are two paragraphs in two contiguous pages that must be merged together after identifying the footnotes N_i and N_{i+1}.


linguistico, ma che la coerenza testuale si esplicita anche grazie a competenze extra-linguistiche. Si veda, *Dizionario di linguistica e di filologia, metrica, retorica*, diretto da G. L. Beccaria, Einaudi, Torino 1996, pp. 721 e sgg.

[18] E. Coseriu, *Linguistica del testo*, cit., pp. 53-60. Coseriu esemplifica l'errata impostazione metodologica di una «linguistica della lingua» ponendo il problema della funzione delle *Modalpartikeln* in tedesco a partire dal racconto *Der Bau* di Kafka.

[19] W. Heinemann, *Aspekte der Textsortendifferenzierung*, in *Text- und Gesprächslinguistik. Handbuch zur Sprach- und Kommunikationswissenschaft*, cit., p. 534. In questo studio Heinemann affronta anche il dibattito intorno alla funzione testuale (pp. 533-536).

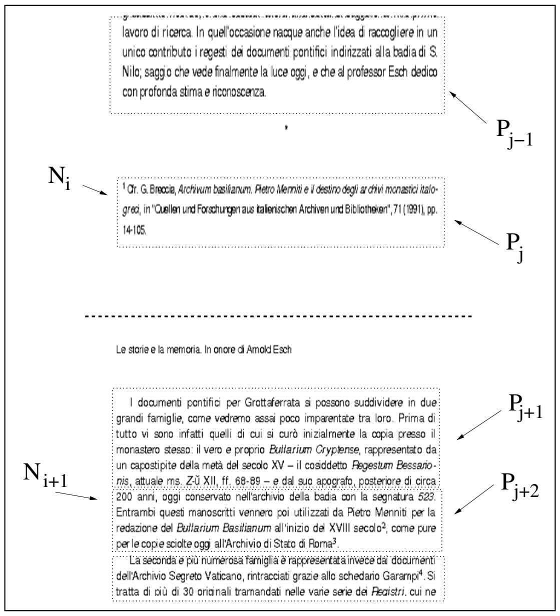
[20] G. Antos, citato da U. Fix, *Textualität und Stil*, in E. Neuland, M. Foschi Albert, M. Hepp (a cura di), *Texte in Sprachforschung und Sprachunterricht*, iudicium Verlag, München 2006, p. 62. Nel suo studio Fix traccia un profilo del recupero dell'aspetto stilistico indicandone la rilevanza per il testo.

[21] Cfr. C. Di Meola, *La linguistica tedesca*, Bulzoni, Roma 2004, p. 187.

[22] Cfr. U. Fix, *Aspekte der Intextualität*, in *Text- und Gesprächslinguistik, Handbuch zur Sprach- und Kommunikationswissenschaft*, cit., 1. Halbband, p. 455.

[23] Si veda M. Soffritti. *Textlinguistik und Texte: Was bestimmt das Tempo der*

Figure 2. Potential notes N_i and N_{i+1} must be merged together.


lavoro di ricerca. In quell'occasione nacque anche l'idea di raccogliere in un unico contributo i regesti dei documenti pontifici indirizzati alla badia di S. Nilo; saggio che vede finalmente la luce oggi, e che al professor Esch dedico con profonda stima e riconoscenza.

*

[1] Cfr. G. Breccia, *Archivum basilianum. Pietro Menniti e il destino degli archivi monastici italo-greci*, in "Quellen und Forschungen aus italienischen Archiven und Bibliotheken", 71 (1991), pp. 14-105.

Le storie e la memoria. In onore di Arnold Esch

I documenti pontifici per Grottaferrata si possono suddividere in due grandi famiglie, come vedremo assai poco imparentate tra loro. Prima di tutto vi sono infatti quelli di cui si curò inizialmente la copia presso il monastero stesso: il vero e proprio *Bullarium Cryptense*, rappresentato da un capostipite della metà del secolo XV – il cosiddetto *Regestum Bessarionis*, attuale ms. Z-ŭ XII, ff. 68-89 – e dal suo apografo, posteriore di circa 200 anni, oggi conservato nell'archivio della badia con la segnatura *523*. Entrambi questi manoscritti vennero poi utilizzati da Pietro Menniti per la redazione del *Bullarium Basilianum* all'inizio del XVIII secolo[2], come pure per le copie sciolte oggi all'Archivio di Stato di Roma[3].

La seconda e più numerosa famiglia è rappresentata invece dai documenti dell'Archivio Segreto Vaticano, rintracciati grazie allo schedario Garampi[4]. Si tratta di più di 30 originali tramandati nelle varie serie dei *Registri*, cui ne

Figure 3. The potential note N_{i+1} is *current text* and must be merged with the paragraph P_{j+1}.

feature is related to the sequence of numbers in $Label$s that are checked for consistency. In this case we verify that the sequence of values (usually disjointed in each chapter) is monotone:

$$0 < N_{i+1} - N_i \leq \delta \qquad (1)$$

where δ is a threshold that allows us to identify the sequence even in case of missing labels. In the current experiments, we fixed $\delta = 5$.

Three main cases are handled by the algorithm when scanning the N list. In the first case **the potential note is one actual note** simply because the potential note satisfies the conditions on the font size and Eq. (1). In the case of footnotes, after processing the note we check if it breaks a paragraph continuing in the next page. In this case we merge the two paragraphs. For example, in Fig. 1 we show two paragraphs (P_{j-1} and P_{j+2}) that are separated by two paragraphs (P_j and P_{j+1}) corresponding to actual notes (N_i and N_{i+1}). P_{j-1} and P_{j+2} belong to different pages, but must be merged.

In the second case **the potential note is a part of one actual note**. The potential note does not satisfy the conditions (e.g. Eq. (1) is not satisfied) and the corresponding paragraph is after one actual note. This can happen when one note spanning more lines contains a number (e.g. a date) at the beginning of one TL. The system in this case merges the two note paragraphs and builds the correct note. After processing the note we check if it breaks a paragraph continuing in the next page. In this case we merge the two paragraphs. One example is shown in Fig. 2.

In the last case **the potential note is a part of a text paragraph**. The potential note does not satisfy the conditions and the corresponding paragraph is not contiguous with other notes. In this case we change the class assigned to the paragraph in *current text* and possibly merge it with the other *current text* paragraph. One example is shown in Fig. 3.

C. Exporting Notes

To reconstruct the notes and the corresponding links in the *ePub* file we compute how many $RefNote$ and $Label$ (i.e. notes) are identified in each chapter. In particular, we compute the following ratio: $\frac{NumRef_i}{NumLab_i}$ $i = 1,..,C$; where $NumLab_i$ is the number of labels (and therefore of notes) in chapter i; $NumRef_i$ is the number of References in chapter i and C is the number of chapters. The system writes the notes at the end of the chapter if:

$$Min \leq \frac{NumRef_i}{NumLab_i} \leq Max$$

Dataset	Books	Links	Precision
FUP	8	294	98.64
AUP	9	619	99.82
Total	17	862	99.41

Table I
PRECISION OF ToC IDENTIFICATION.

Type of notes	Books	Notes	*RefNote* **Rec.**	*RefNote* **FP**	*Note* **Rec.**	*Note* **FP**
A	8	3571	3408	1	3427	9
B	4	361	350	1	337	4
C	2	465	450	0	455	0
M	3	0	-	5	-	1

Table II
RESULTS OF THE IDENTIFICATION OF NOTES.

where $Min = 0.8$ e $Max = 1.2$. In this way it is possible to handle books without notes (where $NumRef_i = 0$ for each chapter i). In this case, some $Label$ found can be due, for instance, to numbered lists. Another case occurs when there are few notes. In this situation, the font statistics are not accurate, in particular if there are more numbered lists than notes.

IV. EXPERIMENTAL EVALUATION

In the experiments reported in this paper, we tested the system on a dataset containing 17 books. Eight books came from the open access collection of the *Firenze University Press (FUP)* whereas the remaining 9 books have been downloaded from the open access collection of *Amsterdam University Press (AUP)*.

In total, there are 862 entries in the *ToC*s to be identified. In Table I we summarize the results obtained for this step. The rare errors are fixed by hand in the GUI before continuing the conversion.

In table II we summarize the results obtained when processing the notes. The books are in this case divided according to the kind of notes: **A**, **B**, and **C** denote notes at the end of the page, chapter, or book, respectively. **M** denotes books without notes. For each category we counted the total number of notes printed in the books, the number of references to notes correctly identified, and the number of notes correctly copied in the output *ePub* file. We also counted the false positives for both items.

False positives in ToC identification are due to accidental positions of keywords in the text regions. For instance if the word "Preface" appears in the Introduction in the first position of one textline then the subsequent paragraphs could be interpreted as a (wrong) preface.

Other problems occur for notes. For instance, in scientific books the exponent in one equation can be confused with one reference to a note. Similarly, one number (e.g. a date) at the beginning of one textline in a note can be confused with a new note. Most of the latter problems have been handled by the consistency check in Eq. 1 but some unfortunate cases are not detected even by fixing a low δ threshold.

V. CONCLUSIONS

In this paper we described a tool designed to convert *PDF* books in a re-flowable XHMTL-based format: the *ePub*. To this purpose, we identify from the *PDF* document two main elements: the Table of Contents and the notes in the text. In the first case we look for the printed Table of Contents and for the corresponding text in the book to extract the navigation information described by the *ToC*. In the second case we identify and process the notes in the book taking into account notes occurring at the end of the page, chapter, or book. In books in Human and Social Sciences the notes are an essential feature extensively used as it is clear also considering the large number of notes in our test collection.

The proposed system is designed to build the *ePub* files from books mostly containing notes. However, we are working on the processing of items more common in scientific and technical documents such as equations and table.

REFERENCES

[1] 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*, 2004, pp. 212–224.

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

[3] E. Oro and M. Ruffolo, "PDF-TREX: An approach for recognizing and extracting tables from pdf documents," in *10th Int.l Conf. on Document Analysis and Recognition*, 2009, pp. 906–910.

[4] S. Marinai, "Metadata extraction from PDF papers for digital library ingest," in *10th Int.l Conf. on Document Analysis and Recognition*, 2009, pp. 251–255.

[5] H. Déjean and J.-L. Meunier, "On tables of contents and how to recognize them," *IJDAR*, vol. 12, no. 1, pp. 1–20, 2009.

[6] X. Lin and Y. Xiong, "Detection and analysis of table of contents based on content association," *IJDAR*, vol. 8, no. 2-3, pp. 132–143, 2006.

[7] A. Doucet, G. Kazai, B. Dresevic, A. Uzelac, B. Radakovic, and N. Todic, "Icdar 2009 book structure extraction competition," in *10th Int.l Conf. on Document Analysis and Recognition*, 2009, pp. 1408–1412.

[8] S. Marinai, E. Marino, and G. Soda, "Table of contents recognition for converting PDF documents in e-book formats," in *Proc. 10th ACM symposium on Document engineering*, ser. DocEng '10, New York, NY, USA, 2010, pp. 73–76.

[9] IDPF, "Epub3 international digital publishing forum," March 2011, `http://idpf.org/epub/30`.