2009 10th International Conference on Document Analysis and Recognition

Metadata Extraction from PDF Papers for Digital Library Ingest

Simone Marinai

Dipartimento di Sistemi e Informatica - Università di Firenze
Via S. Marta, 3 - 50139 - Firenze - Italy
marinai@dsi.unifi.it

Abstract

In this paper we analyze our recent research on the use of document analysis techniques for metadata extraction from PDF papers. We describe a package that is designed to extract basic metadata from these documents. The package is used in combination with a digital library software suite to easily build personal digital libraries. The proposed software is based on a suitable combination of several techniques that include PDF parsing, low level document image processing, and layout analysis. In addition, we use the information gathered from a widely known citation database (DBLP) to assist the tool in the difficult task of author identification. The system is tested on some paper collections selected from recent conference proceedings.

1. Introduction

After several years of massive digitization activities, the main libraries hold now large collections of digitized books and journals. Some of these collections are available in Internet and accessible for free download. Nowadays, most documents distributed by publishers are "digital-born" and the need for retro-conversion of document contents is reduced. However, to perform automatic information extraction from PDF documents the files must be processed so as to identify the relevant metadata. The first applications of computers in libraries were related to the MARC (MAchine Readable Cataloging) standard introduced to define a shared format for library records (e.g. [1]). In the mid 1980s most libraries installed personal computers with the aim of allowing full-text search (through catalog indexes or collections of abstracts) to local users. This service has been afterwards opened to a Web based access, and is now widely available in this form. The information indexed in these catalogs is basically composed by administrative metadata that describe the main features of works (authors, title, publisher, etc.) [1]. The metadata annotation can be a time consuming work, in particular when digital-born works are collected from different sources and need to be grouped in a uniform collection. Several steps are required when dealing with digitized documents in large libraries: create structured master and access copies for the digitized works; generate the required image names; build the technical metadata; create word indexes [4]. However, when the aim is to build small collections of digital-born materials, the metadata annotation can be one of the most important costs and it dominates the others.

This paper deals with the automatic extraction of administrative metadata from PDF documents that are now the standard *de-facto* for documents in digital libraries. Metadata extraction is very useful for the retrospective annotation of digital-born works produced by publishers. Among other activities, the European project OAPEN (Open Access Publishing in European Networks [1]) aims at providing ways of publishing scholarly work in Open Access, enhancing access to monographs in the Humanities and Social Sciences. In this context, we plan to extend some of the techniques described in this paper in order to extract metadata from digital-born electronic journals and monographs.

In the system described in this paper, rather than simply extracting the metadata, we propose a full integration with a *state-of-the-art* digital library software, the Greenstone package. The Greenstone open source digital library software provides tools for organizing information and making it available over the Internet, with a uniform interface to access all documents in a collection. In Greenstone, the structure, organization, and presentation of any particular collection are determined when the collection is set up. This includes the format of documents and how documents should be displayed on screen, metadata sources, browsing facilities to be provided, what full-text search indexes are required, and the presentation of search results.

Greenstone provides a graphical interface for collection

[1] www.oapen.org

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.232

design and construction. People who create digital libraries need to gather documents for inclusion, defining a suitable metadata set, and assigning metadata to each document. Source material is imported into the system through plug-ins that handle various document formats (plain text files, HTML Web pages, Microsoft Office files, PDF documents) [2] [3] [11]. Unfortunately, the PDF plug-in does not handle very well technical papers, and does not automatically extract administrative metadata (see the discussion in Section 2). To deal with this problem we developed the `pdf2gsdl` package to automatically extract administrative metadata from PDF articles.

Portable Document Format (PDF) is designed to allow users to exchange, view, and print electronic documents preserving their look in the most common architectures. From the information extraction point of view one problem is that the order of textual objects in the file does not always correspond to the reading order. Several converters are available either open-source or commercial (a good survey can be found in [9]).More recently, in [8] text objects are extracted from the PDF and a word and line segmentation is produced based on heuristics using the distance between characters and their geometrical positions. A flexible method for detecting and understanding tables in PDF files is discussed in [10]. To parse the PDF file and extract text and graphics objects the PDFBox Java library is used.

Metadata identification is related to the information extraction from objects belonging to digital libraries, that received significant attention in the last few years. Some techniques performed citation analysis in scientific papers. For instance, in [16] each reference in the paper is identified, parsed (to extract a list of author names and publishing year) and corresponding citations in the body of the text are identified. Some papers dealt with the information extraction from *Table of Contents* and *index pages* to improve the indexing of documents in digital libraries [7] [14] [17]. A related problem is the processing of tables to extract suitable metadata for each table. TableSeer [13] crawls various types of documents from Internet (e.g. HTML, PPT, WORD, PDF) to identify tables to be processed. Logical entity recognition in heterogeneous document collections is addressed in [5], where logical and physical features of a segmented document image are represented in an X-Y tree. During the recognition, the layout style and logical entities of an input document are recognized simultaneously by matching the input tree to the trees in the training set. The automatic extraction of bibliographic information from paper titles belongs to the functional labeling tasks in layout analysis. The problem has been addressed, for instance, in [18] where an example-based method to logical labeling of first pages of technical papers is proposed. Given a page segmentation the method finds the best matching document in the database and then transfers the labels of the model to the incoming document. The classes of interest are, among the others, Title, Author, Abstract and Affiliation.

This paper is organized as follows. In Section 2 we analyze the main features of the Greenstone digital library software. In Section 3 we describe the architecture of the software that we designed to extract administrative metadata from PDF articles. In Section 4 we report some experiments that we performed in order to evaluate our system. Some conclusions are then drawn in Section 5.

2 Greenstone

The Greenstone open source digital library software provides tools for organizing information and making it available over the Internet, with a uniform interface to access all documents in a collection. Best suited to build small-medium size libraries, this package is widely diffused and has been used to collect thematic digital libraries to be distributed in CD-ROMS. Due to these features we feel that Greenstone can be used to build personal digital libraries of research papers gathered in Internet or collected from conference proceedings. The limited extent of this kind of library does not justify the burden of a manual annotation of metadata required by a digital library. Rather, we believe that a user would prefer to simply store the papers of interest in a repository and ask the system to automatically include the documents in the collection.

The Greenstone plug-in `metadataPDFPlug`, defined to import PDF documents, is based on the `pdftohtml` software suite, that in turn relies on the PDF reader `xpdf` and the `ghostscript` libraries. The `pdftohtml` tool works well to merge text lines in paragraphs with uniform formatting rules. Text and reading order can be preserved from layout analysis on the XML output of `pdftohtml` by using the top-left bounding box coordinates and sorting first on y then on x [6].

However, this general purpose tool does not handle very well technical papers in Greenstone (see Section 4). One reason is that the actual input of documents in Greenstone is made by means of the *html* plug-in that in turn processes the output of `pdftohtml`. Moreover, the `metadataPDFPlug` does not automatically extract administrative metadata. In principle, it would be possible to design a specific plug-in to solve this problem, however Greenstone plug-ins are wrote in `perl` and it turned out to be not very easy to customize in a suitable way the existing plug-ins. To handle this problem we developed the `pdf2gsdl` package to extract the metadata from PDF papers. We use one robust and standard plug-in (the `metadataXMLPlug`) that allows to import XML metadata and we designed the `pdf2gsdl` package to extract the information from PDF papers and export it into a well formed XML file.

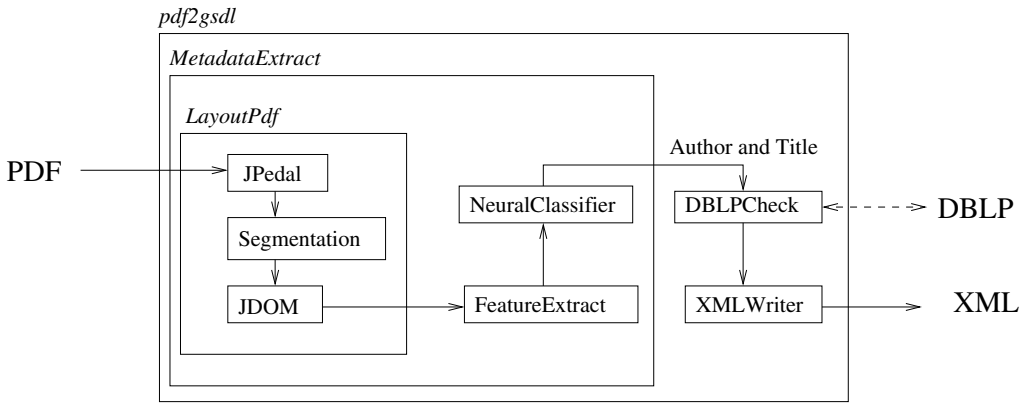

Figure 1. `pdf2gsdl` **architecture.**

3 The pdf2gsdl software

The *pdf2gsdl* is a command line program that allows users to automatically extract administrative metadata from PDF files corresponding to articles. In Figure 1 we show the architecture of the *pdf2gsdl* software. Each PDF file is processed by various modules that are described in the following.

First, the file is opened with the *JPedal* package (*JPedal* is a commercial software, but it includes one library that can be used for non-commercial use) that extracts the basic objects in the page and provides additional information for each textual object, such as its position and font size. By using some simple rule-based heuristics, we extract the page layout with the `LayoutPDF` module that we developed (likewise all the other modules) in Java. This module implements some rules that merge together textual blocks in the horizontal and vertical directions, avoiding to merge separate columns or separate paragraphs. This information is then serialized with *JDOM* so as to obtain an XML tree that contains the features of each region in the page.

The information for each region is processed by the `FeatureExtract` module that represents each region with a numerical feature vector. The feature vector is used as input to a Multi-Layer Perceptron (MLP) classifier [15] to identify regions that could contain the Title and the Authors of the paper. The MLP therefore acts as functional labeler. The features of each region that are considered as input to the classifier can be divided in three main groups. *Graphical features* include the position in the page, the width and height of the region and the page in the document it belongs to (first pages of papers are most likely to contain the administrative metadata we are looking for). *Textual features* include the number of characters, the number of bold or italics characters. *Neighbor features* include the number of neighboring regions and their distance.

The `NeuralClassifier` module assigns scores to regions on the basis of the likelihood that they contain the desired metadata. In order to identify the correct region the text in each region should be "read" to verify its content. We therefore implemented the `DBLPCheck` module that is used to verify the information in the regions consulting the *DBLP* database. At the end of the process, the information related to the metadata is exported in an XML file readable by Greenstone.

3.1 The `DBLPCheck` module

DBLP is a bibliography server that provides bibliographic information on major computer science journals and proceedings. Initially the server was focused on DataBase systems and Logic Programming (DBLP), now it is gradually being expanded toward other fields of computer science [12]. DBLP lists more than a million computer science papers including papers from VLDB, IEEE, ACM and several conference proceedings (including ICDAR, GREC, DAS among the others). The database can be consulted on-line, but it is also possible to download a dump of the database in a *MySQL* format. In the latter case the information is split into three main tables: `dblp_author_ref` contains the author names; `dblp_pub` contains information on the publication (title, publishing date, etc.); `dblp_ref` stores the references between DBLP documents.

After importing the DBLP database in *MySQL*, it is possible to consult it from `DBLPCheck` with suitable JDBC commands. The database is used to check the identified Title/Authors fields. As we will discuss in Section 4, `pdf2gsdl` works well to identify the paper titles, but is less robust when looking for the authors. We can therefore use the `DBLPCheck` module to verify this information. Depending on the DBLP contents it is also possible that the paper is already in the database thus improving the quality

of extracted metadata.

In DBLPCheck we first attempt to locate the input paper in the dblp_pub table by looking for the expected title. Even if simple to define, this task is not trivial, since the title string obtained from MetadataExtract is unlikely to be identical to the entry in the *MySQL* table. We therefore perform several queries on the database so as to identify potential hits with a voting mechanism, looking for words in the title in the dblp_pub table. If a match is found, then the authors are checked and at the end the information from *DBLP* is exported in XML.

In most cases the previous check does not end successfully. This can be due either to a significant difference between the two titles, or to the absence of the paper from *DBLP*. Even in this case the *DBLP* information can be used to improve the recognition of our system (see Section 4 for a numerical evaluation). The idea is that an author could be present in DBLP also with other papers, and also that names/surnames in dblp_author_ref model the universe of most possible authors in *DBLP*. We therefore use the *DBLP* information to identify, by voting, regions that are more likely to contain the authors. To use this approach we perform several queries to the database using the like operator of SQL combined with the words in the field. For instance, with the following query:

```
SELECT * FROM dblp_author_ref
WHERE author like'% "+word[k]+" %'
```

we look for records in the table where the k-th word in the field (WORD[k]) is in any position in the author's field.

4 Experimental Results

To verify the performance of our system we made some experiments on various collections of papers. We summarize in this section the most interesting results.

In the first experiment we considered 80 papers coming from two conference proceedings. 35 papers have been selected from the ICDAR 2003 conference and are printed in IEEE format (on two columns). 45 papers came from the GREC 2003 workshop that is published by Springer Verlag on a single column style. It is important to notice that we designed pdf2gsdl to be independent from any specific publishing format. For all these 80 documents we extracted the title/authors and we inserted the PDF files in Greenstone.

In Table 1 we summarize the number of Title/Author fields that have been identified or not when using the pdf2gsdl software without using the DBLPCheck module. With *Title/Author not found* we mean that no title/author has been detected, or that the detected object is wrong. In the case of *Title/Author + other* the detected title/author contains additional text with respect to the correct one. From the table we can notice that the percentage of titles perfectly identified is very high. Moreover, when the title is found no spurious text is included in the title region. The Author field was more problematic, and this is due to two main reasons. First, frequently the correct field contains some spurious text (affiliation, address, etc.). Second, in the ICDAR case the font of the Author field is the same of the body, and therefore it can be confused with the rest of the paper.

Result	GREC03	ICDAR03
Title found	34	42
Title + other	0	0
Title not found	1	3
Authors found	5	15
Authors + other	24	5
Authors not found	6	25

Table 1. Results on the Title/Author identification for the proceedings of two conferences.

In the second experiment we evaluated the performance of the DBLPCheck module on a larger dataset containing 246 papers published in the ICDAR 2005 proceedings. We first ran the system with the DBLPCheck module activated and we got a 100% recognition, since the conference is indexed in *DBLP*. To better evaluate the system we removed the ICDAR 2005 records from the database and ran again the system looking only for authors. Table 2 compares the results obtained with and without the DBLPCheck module. From the table we can verify that the use of the DBLPCheck module allowed us to increase the percentage of Authors found from 40% to 73,58% of the papers.

Result	DBLPCheck **OFF**	DBLPCheck **ON**
Authors found	75	99
Authors found + other	25	82
Authors not found	140	47
Partial Results	6	18

Table 2. Using DBLP to improve the identification of authors' fields.

In the last experiment we checked the actual import of the papers in Greenstone. To this purpose we manually analyzed the html page that is created by Greenstone for each document.

The results reported in Table 3 have been obtained considering the 80 papers in the first two collections. The errors reported in the table are due to the standard Greenstone's

Result	papers
Perfect	18
File unreadable	2
Images missing	3
Errors in math equations	36
Excessive number of newlines	21

Table 3. Results of the import in the Greenstone system.

PDFPlug and are not related with the software described in this paper. However, it is interesting to notice that in most cases the results are not satisfactory from the user point of view. In particular, tables, equations, captions and pictures are not managed very well and could be addressed by state of the art techniques in document image analysis.

5 Conclusions

In this paper we analyzed `pdf2gsdl`, a software tool that we developed to extract administrative metadata from digital articles. The system exports this information in an XML format that is readable by the Greenstone software. The two software pieces can be used together to easily build a personalized digital library to support research in a specific field. To improve the recognition capabilities of authors, we integrated the system with the information provided by a bibliographical database.

We are currently working on the extension of these techniques to extract additional metadata from digital works in the *Firenze University Press* catalog. The extended packages will be used to prepare XML data associated with monographs and journals to be exported in the OAPEN digital library.

Acknowledgements
We tank S. Giannini, R. Moroni, F. Santanni, and A. Tarocchi for their contribution to the `pdf2gsdl` project.
This work is partially supported by OAPEN EContentPlus Project, co-funded under the EU 7th Framework Programme.

References

[1] W. Y. Arms. *Digital Libraries*. MIT Press, 2000.

[2] D. Bainbridge, K. J. Don, G. R. Buchanan, I. H. Witten, S. Jones, M. Jones, and M. I. Barr. Dynamic digital library construction and configuration. In *Proc. European Conference on Digital Libraries*, pages 1–16, 2004.

[3] D. Bainbridge, J. Thompson, and I. Witten. Assembling and enriching digital library collections. In *Proc. Joint Conference on Digital Libraries*, pages 323–334, 2003.

[4] J. Borbinha, J. Gil, G. Pedrosa, and J. Penas. The case of the digitized works at a national digital library. In *DIAL '06. Second Int'l Conference on Document Image Analysis for Libraries*, pages 116–125, 2006.

[5] S. Chen, S. Mao, and G. Thoma. Simultaneous layout style and logical entity recognition in a heterogeneous collection of documents. In *ICDAR 2007. Ninth Int'l Conf. on Document Analysis and Recognition*, pages 118–122, 2007.

[6] A. Christiansen and D. Lee. Relevance feedback query refinement for PDF medical journal articles. In *CBMS 2006. 19th IEEE Int'l Symposium on Computer-Based Medical Systems*, pages 57–62, 2006.

[7] H. Déjean and J.-L. Meunier. Structuring documents according to their table of contents. In *DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering*, pages 2–9, New York, NY, USA, 2005. ACM.

[8] H. Déjean and J.-L. Meunier. A system for converting PDF documents into structured XML format. In *DAS '06. 7th IAPR Int'l Workshop on Document Analysis Systems*, pages 129–140, 2006.

[9] 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.

[10] 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.

[11] I. H.Witten, K. J. Don, M. Dewsnip, and V. Tablan. Text mining in a digital library. *Int.l Journal on Digital Libraries*, 4:56–59, 2004.

[12] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *SPIRE 2002: Proc. 9th Int'l Symposium on String Processing and Information Retrieval*, pages 1–10, London, UK, 2002. Springer-Verlag.

[13] Y. Liu, K. Bai, P. Mitra, and C. Giles. Searching for tables in digital documents. In *ICDAR 2007. Ninth Int'l Conf. on Document Analysis and Recognition*, pages 934–938, 2007.

[14] S. Mandal, S. Chowdhury, A. Das, and B. Chanda. Detection and segmentation of table of contents and index pages from document images. In *DIAL '06. Second Int'l Conference on Document Image Analysis for Libraries*, pages 70–81, 2006.

[15] S. Marinai, M. Gori, and G. Soda. Artificial neural networks for document analysis and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(1):23–35, 2005.

[16] B. Powley and R. Dale. Evidence-based information extraction for high accuracy citation and author name identification. In *Proceedings RIAO 2007*, pages 62–77, 2007.

[17] P. Sarkar and E. Saund. On the reading of tables of contents. In *DAS '08. Proc. Eighth IAPR Int'l Workshop on Document Analysis Systems*, pages 386–393, 2008.

[18] J. van Beusekom, D. Keysers, F. Shafait, and T. Breuel. Example-based logical labeling of document title page images. In *ICDAR 2007. Ninth Int'l Conf. on Document Analysis and Recognition*, pages 919–923, 2007.