2012 10th IAPR International Workshop on Document Analysis Systems

OCR-Free Table of Contents Detection in Urdu Books

Adnan Ul-Hasan*, Syed Saqib Bukhari*, Faisal Shafait†, and Thomas M. Breuel*

*Department of Computer Science,
Technical University of Kaiserslautern, Germany.
†German Research Center for Artificial Intelligence (DFKI),
Kaiseralutern, Germany.
Email: {adnan,bukhari,tmb}@cs.uni-kl.de, faisal.shafait@dkfi.de

***Abstract*—Table of Contents (ToC) is an integral part of multiple-page documents like books, magazines, etc. Most of the existing techniques use textual similarity for automatically detecting ToC pages. However, such techniques may not be applied for detection of ToC pages in situations where OCR technology is not available, which is indeed true for historical documents and many modern Nabataean (Arabic) and Indic scripts. It is, therefore, necessary to develop tools to navigate through such documents without the use of OCR. This paper reports a preliminary effort to address this challenge. The proposed algorithm has been applied to find Table of Contents (ToC) pages in Urdu books and an overall initial accuracy of 88% has been achieved.**

***Keywords*-Book structure extraction; OCR-free ToC detection; AutoMLP; Urdu document image analysis;**

I. Introduction

Document image understanding is related to the extraction of semantic information from a document. This information may include author names, abstract, table of contents, etc. This understanding can, then, be used for developing applications such as automatic routing of documents, navigation through a scanned book, etc. Table of Contents (ToC) is being used to navigate through documents for ages and it reveals the whole flow of a document to its reader in an elegant way. Automatic detection of ToC pages enables one to navigate through huge volumes of scanned pages efficiently. There has been an increase in interest in document analysis community to develop efficient algorithms for detecting ToC pages in a document [1]. The research related to ToC analysis can be divided into three areas [2]: ToC page detection, ToC parsing and to link the actual pages with these recognized parts.

A quick literature review reveals the presence of various approaches in aforementioned three areas related to ToC extraction. Some researchers have dealt with this problem individually, while others have combined one or more tasks together to make their techniques more robust. Dejéan et al. [3] presented a method for structuring a document according to table of contents, table of figures, etc. They proposed a two-stage solution to this problem, where at first, text similarity for all text segments is computed pairwise and then a list of candidate ToC is selected based on five functional properties. The best candidate is then selected as ToC of document and its references are determined. Dresevic et al. [4] defined a ToC entry to be a single smallest group of words with the same title target somewhere in the book. They considered both ToC page and the target page for reliable ToC detection. Lin et al. [5] treated a ToC as *collection of references to individual articles* and developed an algorithm to associate the contents and page numbers on a ToC page with rest of the pages of the document. Belaïd [6] proposed a Part-of-Speech tagging (PoS) based algorithm for ToC detection. Gao et al. [2] proposed a ToC parsing and decorative elements detection technique based on clustering. Mandal et al. [7] used the spatial distribution properties of contents for detection of ToC pages. They identified ToC pages based on analysis of rightmost words of each line, where words were isolated in a text line based on intra-word spaces. Luo et al. [8] proposed a four-phased algorithm for detecting ToC pages in Japanese books. They defined a ToC page as being composed of three elements: section title, page numbers and connectors. The basic idea is to remove non-ToC pages by identifying some differences between recognized objects and above-mentioned three elements.

All techniques mentioned above for detection of ToC pages (except [7]) share the use of textual similarity (OCR) to detect a ToC page. The use of OCR definitely increases the detection probability of a ToC page. However, a technique based on OCR is bound to fail for documents for which no state-of-the-art OCR is available. This is indeed the case of historical documents in ancient scripts and even modern literature in Nabataean cursive scripts like Persian, Arabic, Urdu[1], etc. and many Indic scripts like Tamil, Bengali, Telugu, etc. Despite the presence of some recent developments in layout analysis systems for Arabic and Urdu documents [9], the non-existence of commercial or open-source OCR techniques for these scripts make it difficult to navigate efficiently through scanned documents. Even the OCR-free technique presented by [7] can not be applied to these scripts due to highly non-uniform distribution of intra and inter word distances [10]. Moreover, lack of knowledge

[1] http://en.wikipedia.org/wiki/Urdu

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.59

about location of the digits would make it impossible to differentiate between a ToC page and a page whose structure is similar to a ToC page.

This paper aims to address the problem of ToC detection by proposing an OCR-free algorithm. The scope of this work is limited to detect ToC pages in Urdu books. Problems of ToC parsing and linking are not addressed in the current paper. This is a preliminary effort in developing OCR-free tools for navigating through historical documents and for other modern scripts where no commercial OCR system is available. Urdu is the native language of over 60 million people in India and Pakistan and it possesses a rich cultural heritage in the form of literature from poets and philosophers. Urdu script is derived from Arabic language and is written from right to left. There is currently no OCR tool (commercial or open-source) available for Urdu Nastaleeq[2] font, in which most of the Urdu documents have been produced.

The proposed algorithm is a mix of machine learning and rule-based techniques and it exploits the specific properties of a typical Urdu ToC page, that is, page numbers are left aligned forming a *column structure* (see Figure 2 for some examples of ToC pages in an Urdu book). This is unlike most of the western scripts where page numbers are most likely right aligned. The presented method consists of two stages, where the binarized document image is first segmented into digits and non-digits using multilayer perceptron (MLP) classifier. The distribution of digits are then investigated in the second stage for presence of a column structure by combining vertical projection profile analysis with run-length encoding. So, the present work may be seen as an improvement of [7] as it uses the knowledge of digits' location in a document as the criterion for ToC detection.

Rest of the paper is organized as follow: Section II describes the proposed method in details, Sections III discusses the experimental evaluation and conclusions are given in Section IV.

II. Method

A gray scale scanned document image is first binarized in a preprocessing step. Digits and non-digits are then extracted using an MLP classifier from this binarized image in the next step. Then, the distribution of digits on a page is estimated using vertical projection profile analysis. A typical ToC page exhibits a vertical peak in the projection profile analysis corresponding to the digit column. The width of this projection profile is then determined to decide whether the input page is a ToC page or not. The details of each step is described in the following sections.

A. Preprocessing

Binarization is the only preprocessing step in the proposed algorithm and it is done using a local thresholding method (Sauvola's method [11]). Fast implementation of this method, described by Shafait et al. [12], has been employed to speed-up the process. Local window size and the k-parameter are two tunable parameters in Sauvola's method and they are dependent on a particular dataset. Local window size is set to 70×70 and k-parameter is set to 0.3 empirically. The binarized document image is then fed to the digit and non-digit classifier.

B. Digit and Non-Digit Segmentation

The segmentation process is an extension to our previous work [13] and the interested reader is referred to it for further details. In [13], the task was to segment the document image into text and non-text regions, however, the main objective in the presented paper is to segment document image into digits and non-digits based on connected component classification. Two main steps of this algorithm, *feature extraction* and *classification*, are described in some details in the following sections for completeness of this paper.

1) Feature Extraction: There are many features that may be extracted from a connected component for MLP learning, however, the raw shape of a connected component itself is an important feature for differentiating a digit from a non-digit. Moreover, the surrounding area of a connected component may also play an important role as well. The connected component with its neighborhood surrounding is referred to as context in this paper. So, the feature vector of a connected component is composed of *shape* and *context* information. Description of both types of features is presented below.

- Shape of connected component: Unlike our previous work of text and non-text segmentation, size information regarding a connected component can not play any role in differentiating digits and non-digits because both of them share same font size and style. However, the shape of a digit and non-digit components may be learned by an MLP classifier. For generating feature vectors, each connected component is rescaled to a fixed window size. This window size is set to 40×40 through empirical evaluation of training data. The size of a shape-based feature vector is 1600.
- Context of connected component: The neighborhood plays an important role in learning a particular connected component as digit or non-digit. The surrounding context area for calculating feature vector is not fixed for all connected components, but it is a function of component's length (l) and height (h). Such that, for each connected component the area of dimensions $5 \times l$ by $2 \times h$ is chosen empirically by keeping a connected component at center for rescaling. Each connected component with its surrounding context area is rescaled to a 40×40 window size for generating context-based feature vector. The size of a context-based feature vector is 1600.

Thus the size of a complete feature vector size is 3200.

[2] http://de.wikipedia.org/wiki/Nastaliq

2) Classification: As mentioned previously, an MLP classifier is used for digit and non-digit classification. Performance of MLP classifier is sensitive to the chosen parameters values. Therefore, the AutoMLP [14], a self-tuning classifier that can automatically adjust learning parameters, has been used. In AutoMLP classifier, a population for MLP classifiers is trained in parallel. For these MLP classifiers, learning parameters are selected from parameter space which has been sampled according to some probabilistic distribution. All of these MLPs are trained for few number of epochs and then half of them are selected for next generation based on better performance. During the training process, the AutoMLP performs internal validation on a portion of training data.

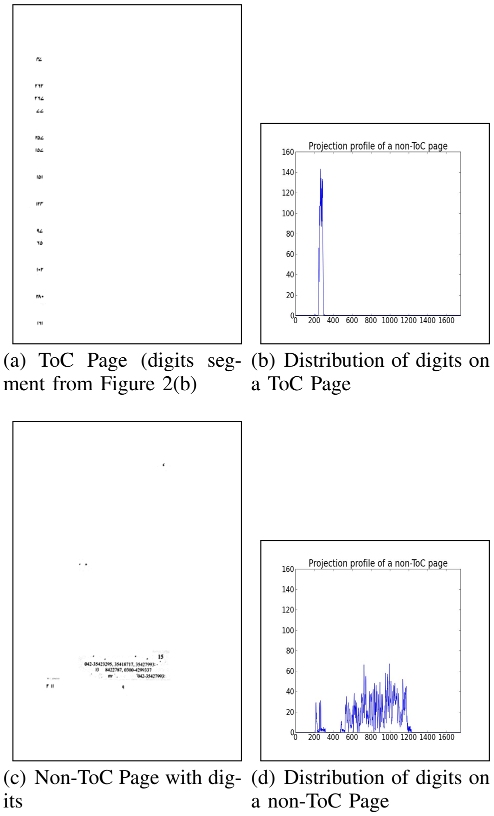
Feature vectors for training AutoMLP classifier have been extracted from dataset of binarized Urdu scanned pages. Due to unavailability of ground-truth information, the digit and non-digit regions are extracted from these pages manually. The size of both digit and non-digit components was increased by using degradation model given in [15]. Around 0.2 million non-digit components and around 0.14 million digit components (containing both Urdu and Latin numerals) are used for training AutoMLP classifier. The reason for including Latin numerals in our training dataset is their common use in Urdu literature, especially in ToC pages. After complete training process, the validation error was around 0.036%.

For testing and evaluation purpose, the feature vector for each connected component of a test document image is extracted in the same way as described in Section II-B1. Then a class label is assigned to each connected component based on classification probabilities of digit and non-digit. Some of the segmentation results are shown in Figure 2.

C. Tabel of Contents (ToC) Detection

After having segmented the document image into digits and non-digits, the next step is to test whether this page is a ToC page or not. The first step to detect a ToC is to draw projection profile of segmented digits on the horizontal axis. The projection profile analysis is the intuitive choice for this purpose as we need to know whether a column structure is present on a page or not. Vertical projection profiles of digits on a sample ToC page is shown in Figure 1(b) and that on a non-ToC page is shown in Figure 1(d). Figures 1(a) and (c) show digit segments of corresponding document images.

It is important to note from Figure 1 that a typical ToC page contains a prominent peak indicating the presence of a column structure. However, a non-ToC page may also contain very short column structure due to presence of digits on a page (see Figure 1(d)). Sometimes, they may be aligned on a page like a ToC structure and may give a false positive. To avoid this situation, a noise threshold (α) has been used, which filters out the column structures of height less than α. The width of columns is then determined using run-lengths over projection profile. However, before final estimation of a column-width, run-length vector is smoothed to fill the gaps between two adjacent non-zero counts of projection profile. The decision on document image being a ToC page is taken on the basis of two thresholds: β and γ, where β is approximately equal or greater than a single digit width and γ ensures the width to be approximately equal to n digits. The value of n is selected as 3 to account for page numbers in the range $1 - 999$. It is also assumed that a ToC page contains no more than two columns of digits to indicate pages numbers. So, we have not considered page as a ToC candidate if it contains more than two columns. This is a fair assumption as a ToC page containing more than two columns for page numbers is quite rare.



(a) ToC Page (digits segment from Figure 2(b)

(b) Distribution of digits on a ToC Page

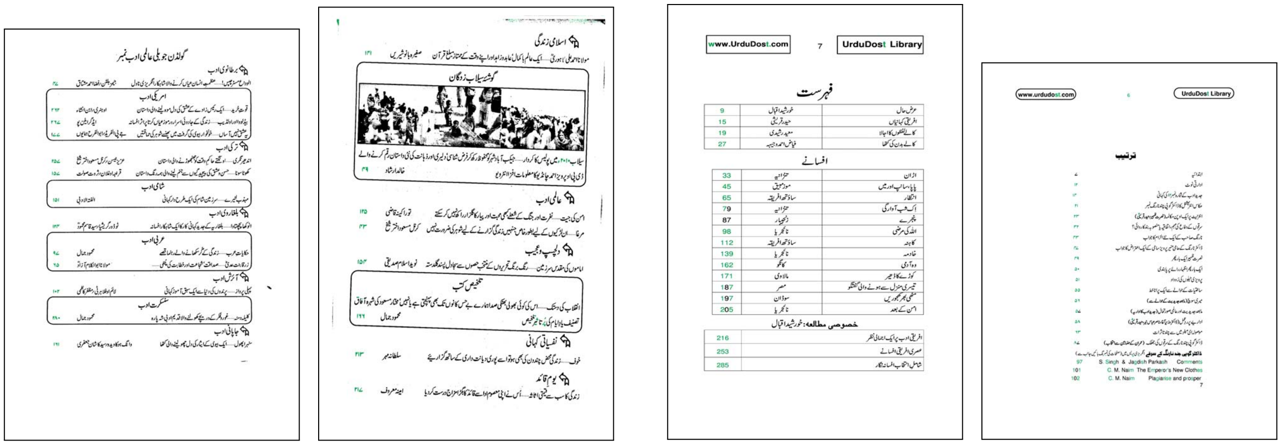
(c) Non-ToC Page with digits

(d) Distribution of digits on a non-ToC Page

Figure 1: Projection profile of digits on a ToC and on a non-ToC Page.

III. Experiments and Results

the The proposed algorithm is evaluated on Urdu digests and books. The digests were scanned at 300 dpi and books were taken from a free on-line Urdu library [16]. There are currently 19 Urdu books and only 2 scanned Urdu digests available in our database. Only one of the books and one scanned digest are used for estimating the threshold parameters described in previous sections and they were not used in testing phase. The parameters α and β are



(a) ToC page of a digest

(b) ToC page with graphics on it

(c) ToC page in a tabular style

(d) ToC page with Urdu and Latin numerals

Figure 2: Some sample ToC pages in the dataset.

evaluated in the ranges of 0.01 to 0.05 and the parameter γ is evaluated in range 0.06 to 0.1 on our training data set, where these ranges are selected on empirical basis for evaluation. Optimal values of these parameters thus estimated are $\alpha = 0.03$, $\beta = 0.01$ and $\gamma = 0.1$. Some sample ToC pages from dataset are shown in Figure 2. It shows that the dataset possesses sufficient variety of typical ToC pages.

The test data of 19 books and 1 digest contain around 3500 pages, out of which 41 pages are ToC. Standard metrics of recall and precision are used to evaluate the performance of ToC detection algorithm. The experimental results are shown in Table- I in the form of a confusion matrix. The proposed algorithm was able to find 36 ToC pages out of 41 ToC pages with an overall accuracy of 88%, a recall of 88% and precision of 69%.

The presented OCR-free ToC detection algorithm is shown to work quite satisfactorily on the Urdu dataset. However, few failure cases need further attention. An important source of error is the misclassification between digits and non-digits which leads to segmentation error (see Figure 3(a)). It is important to note that many Latin alphabets are segmented as numerals as they may not be learned by the classifier. These errors make the detection of column-structure highly improbable. In the case of very short ToC sample, segmentation errors may cause a false negative because of column height below noise threshold, α. These errors can be corrected, though, by improving the digit and non-digit segmentation method.

Table I: Experimental results for ToC detection on Urdu books

	Actual Positive	Actual Negative
Predicted Positive	36	15
Predicted negative	5	3497

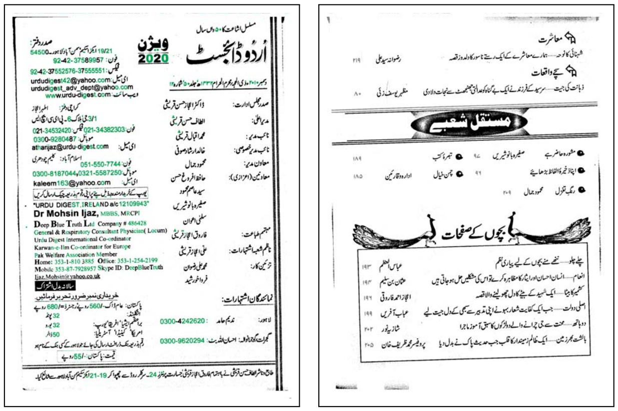
Since this algorithm is based on vertical projection profile analysis, it works fine in absence of document skew. However, it gives false negatives in presence of skew in the document image. This is shown in Figure 3(b). This problem may be solved by an addition step of skew-correction [17] at the preprocessing time.

Moreover, the presented algorithm gives false positive if the document image contains ordinary tables or lists containing digits. A large number of false positives are due to presence of tables containing one column consisting of digits. This problem needs further investigations for finding the specific properties of an ordinary table and a ToC. One possible solution could be to match the digits on ToC page with actual page number it is pointing to. This will, however, need a separate digit-recognizer.

It is also important to note that our method has no post-processing step; however, the detection accuracy may be improved by suitable post-processing step(s). For example, ToC pages are usually present as a group of pages or may be separated by one or two other pages in between, e.g. advertisement pages, etc. So, any page which is far from this group may be rejected in post-processing step. Similarly, ToC pages occur at the start of a book, so, the search for possible ToC pages may be restricted to initial few pages, e.g. first 25%. However, these strategies have not been opted in the current study.

IV. Conclusions

This paper presented a preliminary effort in dealing the problem of navigating through documents in languages where OCR technology is unavailable or not up to state-of-the-art. The proposed algorithm employed machine learning algorithm (AutoMLP) for segmenting the document image into digits and non-digits. Then, the vertical projection



(a) Poor digit/non-digit segmentation

(b) A skewed ToC page

Figure 3: Representative Failure Cases.

profile analysis is employed to detect the column structure of a typical ToC page. Some failure cases are also discussed along with their root causes and remedies. The proposed method may be easily extended for both simple (Latin) and complex (Tamil, Indic, etc.) scripts by training AutoMLP classifier for only digits in a particular language. The current focus of this research work was to develop a basic algorithm for ToC page detection in a multi-page document and further research is needed to develop OCR-free methods for linking the detected ToC entries with corresponding content pages. Application of such algorithms may include on-line and/or off-line digital libraries of cursive scripts, such as Harvard Islamic Heritage Project [18] and free on-line Urdu library [16].

References

[1] A. Doucet, G. Kazai, B. Dresevic, A. Uzelac, B. Radakovic, and N. Todic, "Setting up a Competition Framework for the evaluation of Structure Extraction from OCR-ed Books," *Int. Journal on Document Analysis and Recognition*, no. 14, pp. 45–52, 2011.

[2] L. Gao, Z. Tang, X. Tao, and Y. Chu, "Analysis of Books Docuemnts' Table of Content Based on Clustering," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, Jul. 2009, pp. 911–915.

[3] H. Dejaéan and J.-L. Muenier, "On Tables of Contents and How to Recognize Them," *Int. Journal on Document Analysis and Recognition*, no. 12, pp. 1–20, 2009.

[4] B. Dresevic, A. Uzelac, B. Radakovic, and N. Todic, "Book Layout Analysis: TOC Structure Extraction Engine," *Advances in Focus Retrieval*, pp. 164–171, 2009.

[5] X. Lin and Y. Xiong, "Detection and Analysis of Table of Contents Based on Content Association," *Int. Journal on Document Analysis and Recognition*, vol. 8, no. 2-3, pp. 132–143, 2006.

[6] A. Belaïd, "Recognition of Table of Contents for Electronic Library Cosulting," *Int. Journal on Document Analysis and Recognition*, vol. 4, pp. 35–45, 2001.

[7] S. Mandal, S. P. Chowdhury, A. K. Das, and B. Chanda, "Automated Detection and Segmentation of Table of Contents Page and Index Pages from Document Images," in *12th International Conference on Image Analysis and Processing*, Sep. 2003, pp. 213–218.

[8] Q. Luo, T. Watanabe, and T. Nakayama, "Identifying Contents Page of Documents," in *13th International Conference on Pattern Recognition*, vol. 3, Aug. 1996, pp. 696–700.

[9] S. S. Bukhari, F. Shafait, and T. M. Breuel, "High Perfomance Layout Analysis of Arabic and Urdu Document Images," in *11th International Conference on Document Analysis and Recognition*, Bejing, China, Sep. 2011, pp. 1375–1279.

[10] A. Abidi, I. Siddiqi, and K. Khurshid, "Towards Searchable Digital Urdu Libraries - A Word Spotting Based Retrieval Approach," in *11th International Conference on Document Analysis and Recognition*, Bejing, China, Sep. 2011, pp. 1344–1348.

[11] J. Sauvola and M. Pietikäinen, "Adaptive Document Image Binarization," *Pattern Recognition*, vol. 33, pp. 225–236, 2000.

[12] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images," in *15th Document Recognition and Retrieval Conference (DRR-2008), part of the IS&T/SPIE International Symposium on Electronic Imaging*, vol. 6815. San Jose, CA, USA: SPIE, Jan. 2008.

[13] S. S. Bukhari, M. Al-Azawi, F. Shafait, and T. M. Breuel, "Document Image Segmentation using Discriminative Learning over Connected Components," in *9th IAPR Workshop on Document Analysis Systems*. Boston,, MA, United States: ACM, Jun. 2010, pp. 183–189.

[14] T. M. Breuel and F. Shafait, "Automlp: Simple, Effective, Fully Automated Learning Rate and Size Adjustment," in *The Learning Workshop. The Learning Workshop, March 6 - April 9, Cliff Lodge, Snowbird,, Utah, United States*. Snowbird, Utah: Online, Apr. 2010, extended Abstract.

[15] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuetzle, and D. Madigan, "Document degradation models: Parameter estimation and model validation," in *IAPR Workshop on Machine Vision Application*, Kawasaki, Japan, Dec. 1994, pp. 552–557.

[16] "Urdu free on-line library." [Online]. Available: http://www.urdudost.com/index.php

[17] J. van Beusekom, F. Shafait, and T. M. Breuel, "Combined Orientation and Skew Detection using Geometric Text-Line Modeling," *Int. Journal on Document Analysis and Recognition*, vol. 12, no. 02, pp. 79–92, 2010.

[18] "Harvard Islamic Heritage Project." [Online]. Available: http://ocp.hul.harvard.edu/ihp/