Content-level Annotation of Large Collection of Printed Document Images

C. V. Jawahar and Anand Kumar
Center for Visual Information Technology
International Institute of Information Technology, Hyderabad - 500032, INDIA
{jawahar@, anandkumar@research.}iiit.ac.in

Abstract

A large annotated corpus is critical to the development of robust optical character recognizers (OCRs). However, creation of annotated corpora is a tedious task. It is laborious, especially when the annotation is at the character level. In this paper, we propose an efficient hierarchical approach for annotation of large collection of printed document images. We align document images with independently keyed-in text. The method is model-driven and is intended to annotate large collection of documents, scanned in three different resolutions, at character level. We employ an XML representation for storage of the annotation information. APIs are provided for access at content level for easy use in training and evaluation of OCRs and other document understanding tasks.

1. Introduction

Annotated data is a prerequisite for the development, evaluation, performance enhancement and benchmarking of data driven document analysis systems. Lack of linguistic resources in the form of annotated datasets has been one of the hurdles in building robust document understanding systems for Indian languages. Documents need to be annotated at the structural, functional and content-level for building a dataset for variety of document understanding systems. Content-level annotation is critical for developing recognizers. Meta information in the form of script, language, print/scan parameters etc. are essential for a different set of tasks. In this work, we focus on tools and algorithms required for the annotation of printed documents (primarily from books, partly from magazines and newspapers). The emphasis is on building a large corpus of annotated data for developing robust OCRs, which are not available for Indian languages.

Annotation of data has become central to the success of supervised and semi-supervised machine learning algorithms. The problem of alignment of parallel handwritten and text corpora has been addressed in the context of segmentation of text lines to words by Zimmerman and Bunke [8]. Kornfield *et al.* [4] proposed an approach in which word images are matched to text based on global properties of the words extracted from both the handwritten word images as well as text words that are rendered using a specific font. Elliman *et al.* [3] presented annotation of documents containing cursive writing. Their data set consisted of around 900 sheets of cursive writing, annotated at word level. In a significant work for Indian languages, Setlur *et al.* [9] designed a truthing tool for annotation of words from document images of *Devanagari* scripts. Around 120,000 words from 400 scanned documents were annotated using the tool. For the languages with availability of reasonably good commercial OCRs, annotation has been focused primarily on functional and structural layout information. University of Washington data sets UW-I and UW-II [5] consist of 1771 English and 477 Japanese document images from scientific and technical journals. These datasets store text zone bounds, ground truth data for each zone, finer attributes and qualitative information useful for document image understanding tasks. Japanese Character Image Database [6] contains approximately 180,000 (0.18M) Kanji, Hiragana, Katakana, alphanumeric, and symbolic characters, extracted from a variety of machine printed documents, with quality varying from clean to degraded. Indian languages, with complex scripts and poor commercial systems, immediately need annotated data at the character (*akshara*) level. Annotation of Indian language documents is difficult due to complexity of the scripts. Languages like Gujarati, Kannada, Telugu, Malayalam and Tamil have multiple components in a character, whereas in Hindi and Bangla, all the characters of a word are typically connected together. Our work is towards annotation of large number (Millions) of *aksharas* from multiple Indian language document images. Such a large scale annotation needs efficient algorithms and tools.

Innovative solutions are required to complete this annotation in limited time. We provide structural and functional annotation to the text and graphics blocks using a semi-

automatic tool. We then align image and textual content parallely at word level (Section 3). A model-based annotation, on the lines of [2] is then employed for annotation at the character level. A set of tools with efficient algorithms in the back end achieve this (Section 5).

2. Document image annotation

Document image annotation is the process of labeling image components (often with text) with additional details such as layout information, language or script, details of the print and print conditions etc. The environmental settings and conditions of image acquisition make impact on the quality of images obtained and the applications that use the annotated data. Annotation of large database of document images involves well structured processing, labeling procedures and data storage for easy access and use. The proposed annotation tools are mainly designed for generating datasets for Indian language OCRs. These tools do not assume very complex layout documents.

Most of our scanned documents come out of books. Documents are scanned at multiple resolutions – 200dpi, 300dpi and 600dpi. The original scanned document images are preprocessed by removing noise, correcting skew and thresholding. A copy of every image after each preprocessing step is stored separately. The functional parameters of preprocessing, like skew angle, are required for automatic performance evaluation. Scanned documents are stored in a consistent directory structure. Textual content needed for annotation is typed separately page by page and stored separately. A standard word processor is used for this. Textual content is obtained in UNICODE for the content-level annotation of blocks.

Document images are segmented and functional tags are attached with the help of an easy to use tool. The document image is annotated at multiple structural levels. The segmented blocks are classified as text and non-text blocks. The non-text blocks are labeled as pictures, graphs, mathematical equations or tables. They are not further annotated in this phase of the project. The paragraph blocks are then aligned with the corresponding text for labeling. The paragraphs are again segmented into lines and words, which are labeled with corresponding textual content. The labeling task is direct at the higher level blocks and is done either automatically or semi-automatically. Thus with minimal manual intervention, annotation at word level (Figure 2) is obtained.

For training, testing and bootstrapping OCRs, one needs annotation at *akshara* level. *Akshara* is a fundamental unit (equivalent to character) of Indian language scripts. It could comprise of a single consonant(C), consonant+vowel(CV), CCV or even CCCV. An *akshara* has a single pictorial representation (even a single connected component) and may

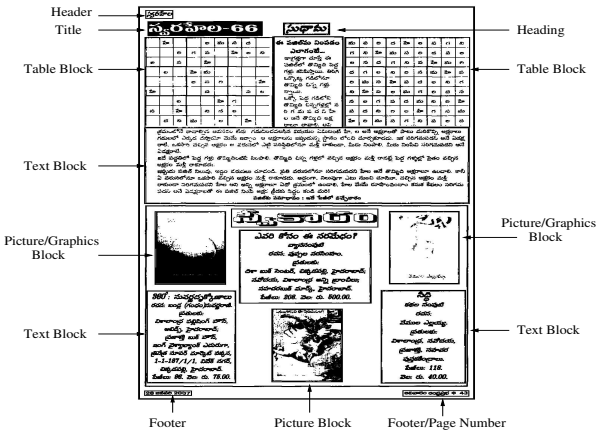

Figure 1. Document image: Blocks of document image identified as text paragraphs, tables and picture/graphics blocks with additional appropriate tags attached.

correspond to one or more UNICODEs. We propagate the annotation from word to *aksharas* using a technique similar to [2] (Figure 2). For simplifying the pattern classification problem in OCRs, an *akshara* may further be split into visual symbols. However, there is no well accepted representation below this unit. A similar model-based annotation could be directly used to obtain lower level annotation.

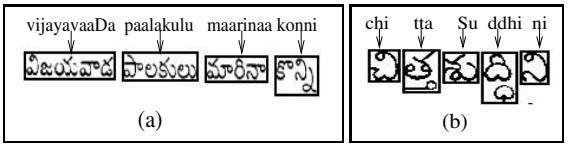

Figure 2. Labeling components of lines and words with corresponding text (in ITRANS): (a) Words of a line. (b) *Akshara* **components of a word.**

Structured schema for storage of annotated data is another important aspect of the annotation. Creation of annotated datasets for future research requires a standard representation for annotation that supports the following requirements: (i) script and content independence (ii) semantic interpretation of the print at various user defined logical levels (e.g. word, character) (iii) capture of information about script, print quality (subjective), font type, style and size (iv) capture of information about data capture environment (vi) separation of scanned printed data from its semantic interpretations.

3. Hierarchical annotation of text blocks

Document image annotation is carried out in a hierarchy of levels. Figure 3 shows the block digram of the annotation process. Given a parallel corpus of document image and corresponding text, the objective is to align the text and image components. Annotation starts at block level followed by line level, word level and ends at *akshara* level.

Text block annotation: In block level annotation, the text and non-text blocks of the document image are classified and labeled (Section 2). The text blocks are segmented into lines and text lines are extracted from their labels. Line level annotation is done by labeling the lines from the paragraph of the document image with their corresponding extracted text lines. Line images are labeled by parallel alignment with the text lines. Images of line segments are again divided into words for word level annotation. Words are obtained from the labels of the images of lines. The word images are labeled with the extracted text words by alignment.

While keying in the unaligned parallel text, the following are specially kept in mind: (i) paragraphs are separated by one or more blank lines. (ii) lines end exactly the same manner as they appear in the image/document (iii) words are separated by blank spaces. This makes the segmentation of text lines trivial and alignment straight forward. However due to errors in the image segmentation and noise in the textual content (spell-errors), the method can fail and user interaction may be required. The detection and correction of segmentation errors, can be modeled as an alignment problem.

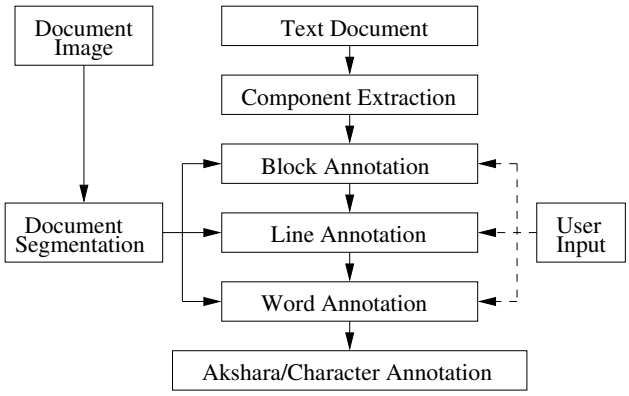

Figure 3. Hierarchical Annotation: Levels of document segmentation and labeling.

Akshara level annotation: *Akshara* annotation is the process of mapping a sequence of connected components from the word image to the corresponding text *akshara*. Image of *akshara* text is rendered for matching with components of the word image. We use the connected component matching module, which is similar to stroke matching module of [2], to come up with the best alignment of the *akshara* components to the corresponding *aksharas* of words. Figure 4 shows the process of *akshara* level annotation. We assume that each *akshara* in the text corresponds to one or more connected components in the word image. However, computing the best assignment of connected components to *aksharas* is not trivial as multiple components can form an *akshara*.

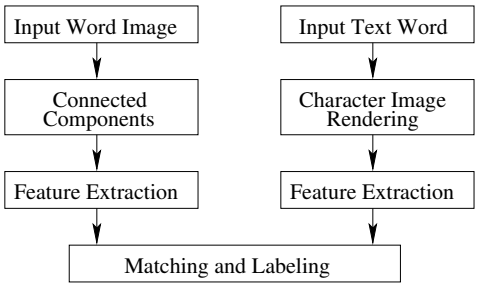

Figure 4. *Akshara* **level annotation of words.**

We employ a modified version of the elastic matching or dynamic time warping (DTW) algorithm to solve this problem. The alignment cost between two sets of connected components is measured by computing distance between a set of features extracted from them. The total cost of DTW is used as a similarity measure to group together components that are related to their root *akshara* by partial match. Elastic matching is able to absorb the possible splits and breaks in the image segmentation. When components of two *aksharas* are merged, the algorithm assigns two *aksharas* as label, which can be corrected using a validation tool. With the assumption that multiple connected components might map to a single *akshara*, we formulate the problem as follows: Find the best alignment between components and *aksharas* using DTW matching score as the cost of grouping multiple components to form the *akshara*. The proposed method avoids errors in segmentation of *aksharas* with overlapping boundaries by extracting *aksharas* from connected components of word image. The merged *aksharas* are annotated with corresponding group of text labels.

Validation of automatic annotation is critical to the correctness of the data. Performance measure of the system under evaluation depends on the correctness of the annotated data. Therefore, automatic annotation of *akshara* is verified and validated using a validation tool. The validation tool provides user interface (UI) to check the correctness of annotated data and validate. The tool provides options to verify and validate similar *aksharas* at once, reducing overhead of doing it for every *akshara* within every document.

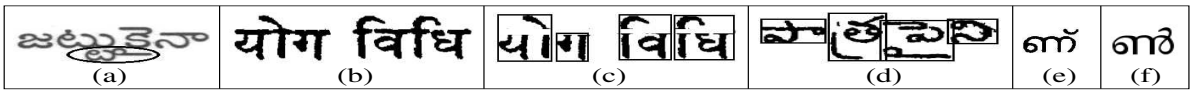

Figure 5. Indic script issues: (a) Touching components of two *aksharas* of Telugu due to formatting errors in word processors, (b) Hindi words with *Shirorekha*, (c) Multiple components after *Shirorekha* removal and (d) Overlapping bounding boxes. Unavailable UNICODE representations for (e) and (f).

4. Annotation complexity of Indic scripts

Manual annotation of document images at *akshara*-level is a labor and time intensive task. The task becomes more challenging when the documents are noisy and degraded. The proposed method annotates document images from the topmost level of blocks down till the *akshara* level. It stores the layout and content information of the document images. Segmentation of many of these documents into blocks and components at every level and the direct annotation is not trivial due to the complexity of the Indic scripts. When the segmentation algorithms fail, user intervention is asked for. Some languages like Hindi and Bangla have special properties (*Shirorekha*) that can be used for effective word level segmentation. Different algorithms are used in the tool to segment Indic scripts.

Since we align parallel text with the word image, we need to create the textual content first. This is done in UNICODE using a standard word processor. Image rendering routines are required to render words for automatic annotation at *akshara* level. UNICODE representation of the *aksharas* is required for rendering an image corresponding to the text. Examples of the complexities of Indian scripts are briefly mentioned below, and demonstrated in Figure 5.

- *Segmentation:* An *akshara* could comprise of multiple connected components or single connected component could be part of multiple *aksharas*. In such cases, segmentation of word into *aksharas* become difficult.

- *Representation of components:* Popular model for representation of components in document image is with the help of bounding boxes. In presence of overlapping bounding boxes it necessitates adding extra information into the representation (for example, the rank of the component of interest). Characters in languages like Hindi, Bangla etc. are connected by a headline called *Shirorekha*. It needs special representation for annotation.

- *Representation of code/text:* Some of the *aksharas* of Indian language scripts do not have UNICODE representation. An example from Malayalam is shown in Figure 5.

5. Annotation, storage and access

User friendly tools are developed for efficient annotation of large corpus of Indian language documents. Tools are developed keeping the following aspects in mind. (i) Easy document browsing and multiple file format support. (ii) Manual options for segmentation and editing. (iii) Ease of annotation and display of hierarchical annotation and (iv) Support for meta data entry. Figure 6 shows screen shot of one of the tools. Large annotated datasets call for a standard representation that is independent of scripts and allow semantic interpretation of the print at various user-defined logical levels. The annotated data is represented in a hierarchy of XML tags, on the lines of [1, 7]. The meta data of the source and actual annotation data are stored separately for easier updation to the schema in future.

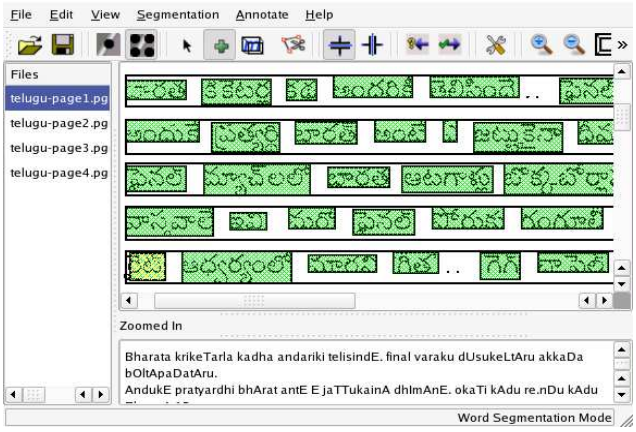

Figure 6. Annotation Tool: Semi-automatic block to word level annotation tool.

Meta data: The meta data captures information about the document sources and data capture environment. It comprises of (i) Title, author, publisher, year of publication and ISBN etc. source specific information (ii) OCR specific data containing font type, style, size and subjective definition of degradation quality etc. (iii) Digitization information comprising of scanner related information, scan resolution used, image file format (compression information) etc.

Annotation data: Bounding box and UNICODE text labels are stored in the schema for all levels of annotation. Wherever bounding box information is ambiguous, additional information is added to attach the tag. Annotations corresponding to *aksharas* is stored as either components or bounding box depending upon the complexity of scripts.

A set of standard application program interfaces (APIs) are designed to access the data from such organized schema. Using these APIs, annotation information at all levels and meta information of the sources can be accessed directly. Figure 7 shows some annotations of Telugu words. The data was transcribed using UNICODE encoding to form a parallel text corpus. The document images were segmented and annotated automatically till word level by using annotation tools developed as explained in section 3.

The *aksharas* from input text word are converted into images by rendering. Any image processing library (eg. ImageMagick, GIMP) would render image of text, given font information. The rendered *aksharas* are then aligned with connected components of the original input word image and the annotations are propagated from the rendered *aksharas* to the character components in the original input word image. The results of word level annotation and propagation to *akshara* are shown in Figures 2 and 7.

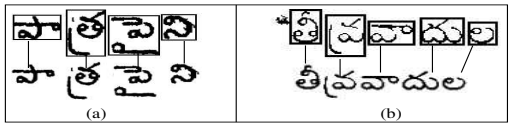

Figure 7. *Akshara* **annotation through matching. Rendered image shown without bounding box. Good component matching in (a) clean images (b) degraded images.**

The algorithm takes care of spurious connected components due to noise, ink blobs etc. that might occur in the word image. The matching process effectively removes it from getting into the annotation. However, when the spurious components are joined with the basic *akshara*, it will be included in the final annotation to provide realistic content-level annotation. When components from two *aksharas* are joined in word images, the algorithm assigns two *aksharas* as label in the final annotation, which can be corrected using validation tool. Figure 7(b) shows a successful example of *akshara* matching in presence of a spurious component in the word due to noise or degradation. It can be seen that the spurious component is separated from the actual *aksharas* in the final annotation. Also, the *akshara* components are grouped together in presence of cuts.

Automatic *akshara* annotation removes significant labor involved in the process, which would be very high if every akshara is annotated manually. Validation of the annotation is very important to ensure correctness. The annotation is verified and validated using another tool. It automatically displays all similar *aksharas* in the UI and allows the quality assurance person to verify and validate them. Thus, verification speed is improved. With the support of good algorithms (like segmentation etc.) in the tools, the annotation becomes more automatic. We are employing the tools and schemes discussed in this paper to annotate around 50 Million *aksharas* across five major Indic scripts.

6. Conclusion and future work

We proposed an approach for hierarchical annotation of document images on a very large scale. The annotation is carried out in a number of steps using automatic and semi-automatic annotation and validation tools. The tools are developed, tested and annotation is progressing.

Acknowledgments: Authors thank MCIT, Government of India for the financial support. Authors also thank members of the consortium working on development of Indian language OCRs for their inputs and cooperation.

References

[1] A. Bhaskarbhatla, S. Madhavanath, M. Pavan Kumar, A. Balasubramanian, and C. V. Jawahar. Representation and annotation of online handwritten data. In *Proc. of 9th IWFHR*, pages 136–141, 2004.

[2] Anand Kumar, A. Balasubramanian, Anoop M Namboodiri, and C.V. Jawahar. Model-based annotation of online handwritten datasets. In *Proc. of 10th IWFHR*, 2006.

[3] D. Elliman and N. Sherkat. A truthing tool for generating a database of cursive words. In *Proc. of 6th ICDAR*, pages 1255–1262, 2001.

[4] E.M. Kornfield, R. Manmatha, and J. Allan. Text alignment with handwritten documents. In *Proc. of Int. Workshop on Document Image Analysis for Libraries*, pages 195–209, 2004.

[5] I. Guyon, R. Haralick, J. Hull, and I. Phillips. Data sets for OCR and document image understanding research. In *Handbook of Character Recognition and Document Image Analysis*, pages 779–799, 1997.

[6] Japanese Character Image Database. *The Center of Excellence for Document Analysis and Recognition, State University of New York at Buffalo*.

[7] M. Agrawal, K. Bali, S. Madhvanath, and L. Vuurpijl. UPX: A new XML representation for annotated datasets of online handwriting data. In *Proc. ICDAR*, pages 1161–1165, 2005.

[8] M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line database for handwritten English text. In *Proc. of 16th ICPR*, pages 35–39, 2000.

[9] S. Setlur, S. Kompalli, V. Ramanaprasad, and V. Govindaraju. Creation of data resources and design of an evaluation test bed for devanagari script recognition. *Int. Workshop on Research Issues in Data Engineering: Multi-lingual Information Management*, pages 55–61, 2003.