A tool that converted 200 Tamil books for use by blind students

Shiva Kumar H R and A G Ramakrishnan
Department of Electrical Engineering, Indian Institute of Science, Bangalore.

Abstract

A versatile tool has been created, with user-friendly interface, for the rapid and efficient conversion of printed Tamil books to Braille books for the use of persons with visual disability. The tool has been developed in Java using Eclipse SWT and runs on Linux, Windows and Mac operating systems. This tool has been developed as an open source project and is available under the Apache 2.0 license from code.google.com. An individual scanned page or all the pages of a whole book can be recognized by this tool. The average time taken for digitizing a Tamil page is two seconds. The output can be saved in RTF, XML or BRF (Braille) format directly, by the click of a button. There is a provision for manually selecting the individual columns of a two-column printed page or even marking the individual rectangular text blocks of a page with a more complex Manhattan layout. The user can modify the reading order of the so-selected text blocks. This information of ordered text blocks is passed on to the Tamil OCR integrated at the backend of the tool and hence the recognized Tamil text in Unicode is put together in the same reading order. In the case of books with identical or very similar text layouts across its pages, such an user-defined layout can be saved as a custom layout and automatically applied to segment the other pages of the same book or a different book also.

Keywords: Tamil OCR, Braille conversion, digitization, column layout marking, automated error detection, error correction, text block separation, line segmentation correction, XML format.

Introduction

In the past two years, when our Tamil OCR was being used by Worth Trust, Chennai on a regular basis, we conducted a study on the work flow and the time taken for the various steps involved in converting the printed book to a Braille book. We realized that while it takes only a few minutes to OCR the whole book, it takes two to three days to correct the OCR errors, convert the Unicode to Braille text and reformat it to the Braille printer and page requirements. This incited us to take upon ourselves the task of developing a comprehensive tool that handles all the issues involved in the work flow and thus reduce the overall processing time from a week to a single day.

On account of this, the tool offers a convenient graphical user interface, as shown in Fig. 1, for effecting corrections of the recognized text. The input image and the output text are displayed side by side in two different windows, which can be simultaneously zoomed in

and out. This results in alignment of text lines between the source image and the OCR output text. Our Indic Keyboard interface [5], again available under Apache 2.0 licence from code.google.com, can be used to edit the Tamil text before saving.

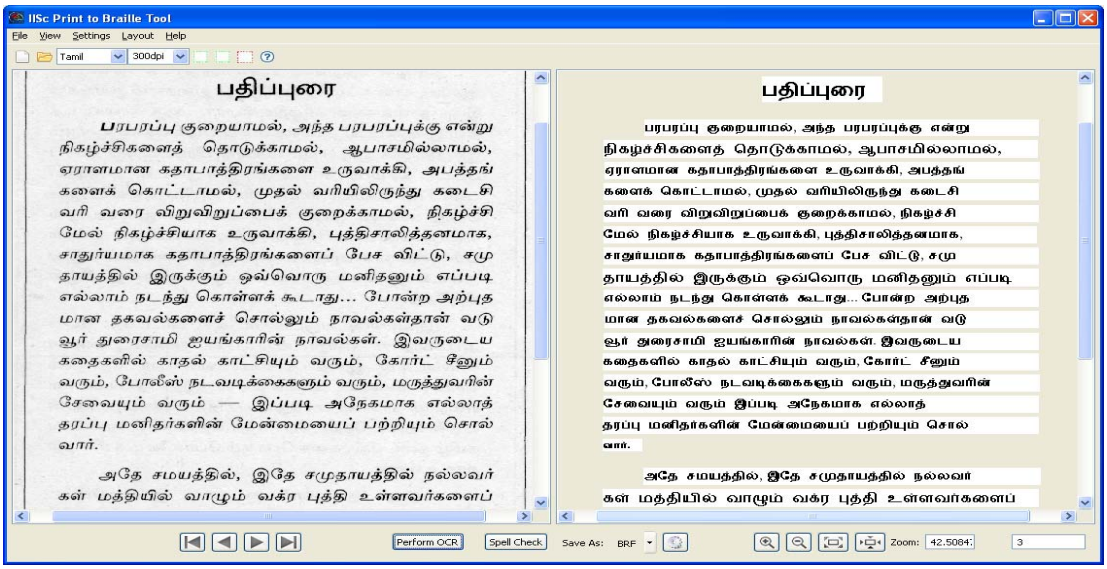

Fig. 1. Aligned display of input image and output text for easy verification and editing.

Output Edit Facility

Using the OCR that has been interfaced, the text check and edit facility tries to identify the wrongly recognized words and suggests a list of alternatives to select from, as well as provisions to edit the recognized word or type a completely new word. The suggested new word

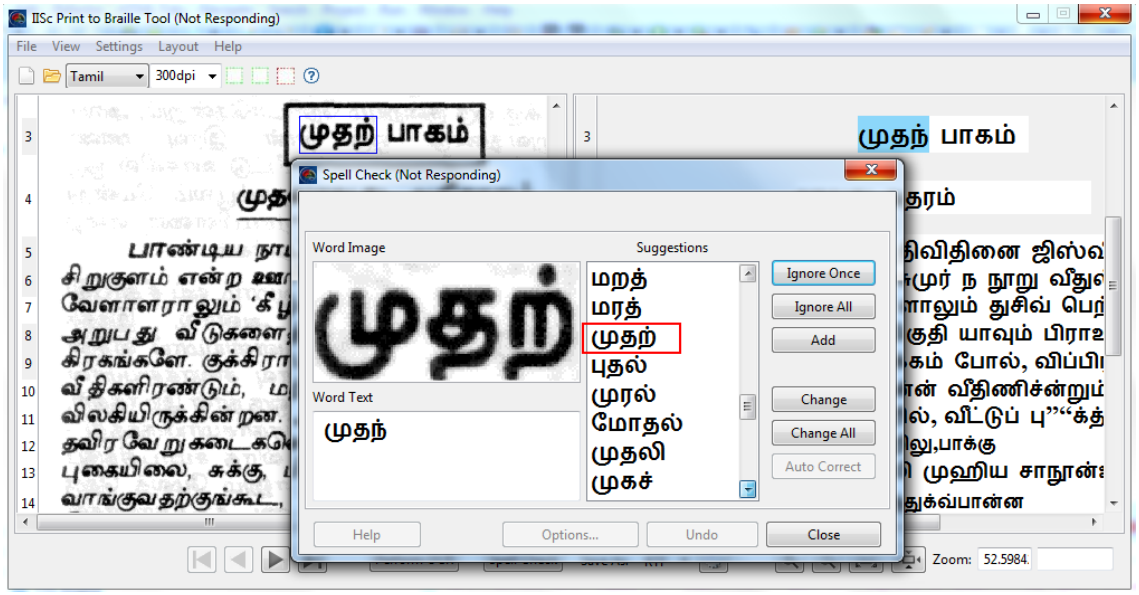

Fig. 2. GUI to correct recognition errors with suggested list of alternatives, etc.

can be used to replace the current instance of the wrongly recognized word or all instances occurring in the entire book.

Correction of logical layout detection

Figure 3 shows the GUI for selecting one of the standard logical layouts for the printed page to be recognized. The user can edit the wrong layout segmentation of the OCR or the standard layout applied by the tool itself. This facility is illustracted in Fig. 4.

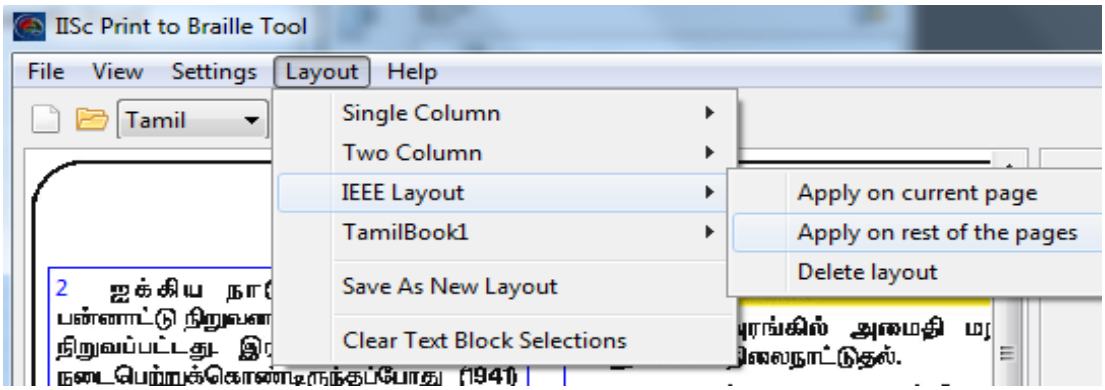

Fig. 3. Tool supports user correction of layout segmentation by the OCR.

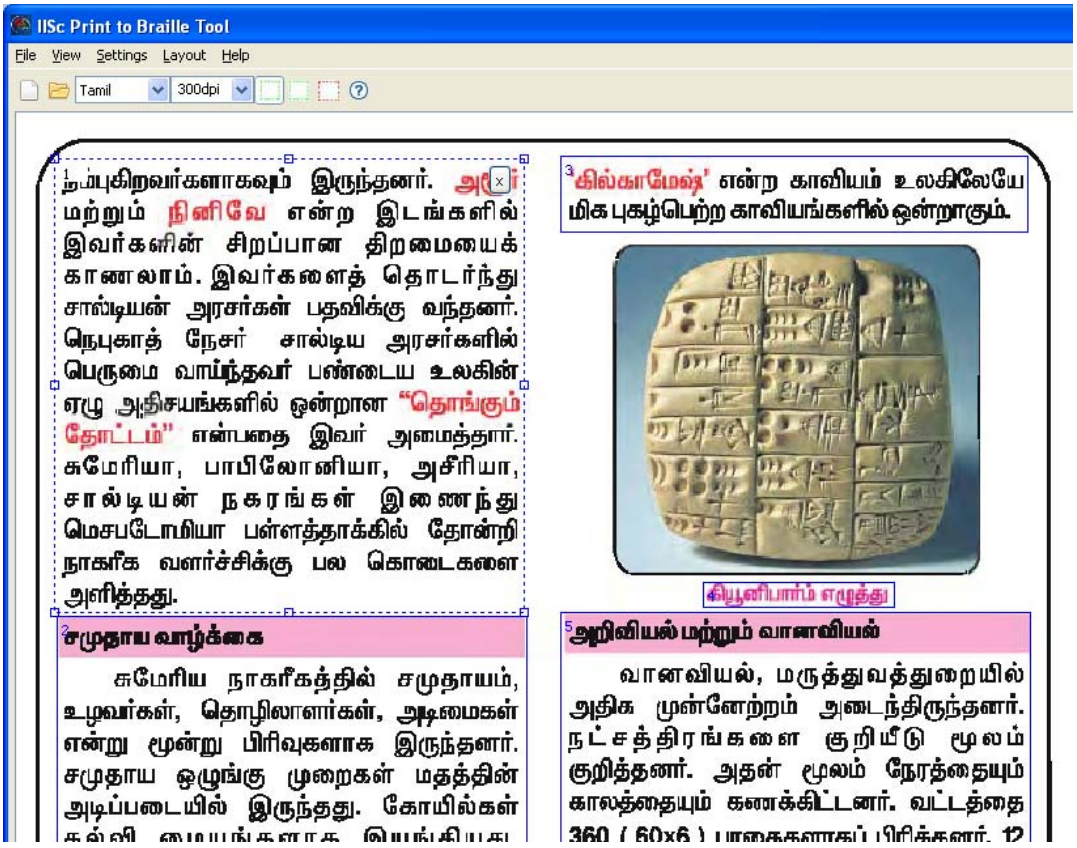

Fig. 4. Facility to mark and edit text blocks and their logical order.

Correcting Line Segmentation Issues

Similarly, if the OCR wrongly segments the text lines, the user can rectify the under or over segmentation of the text lines, using the convenient interface available, shown in Fig. 5.

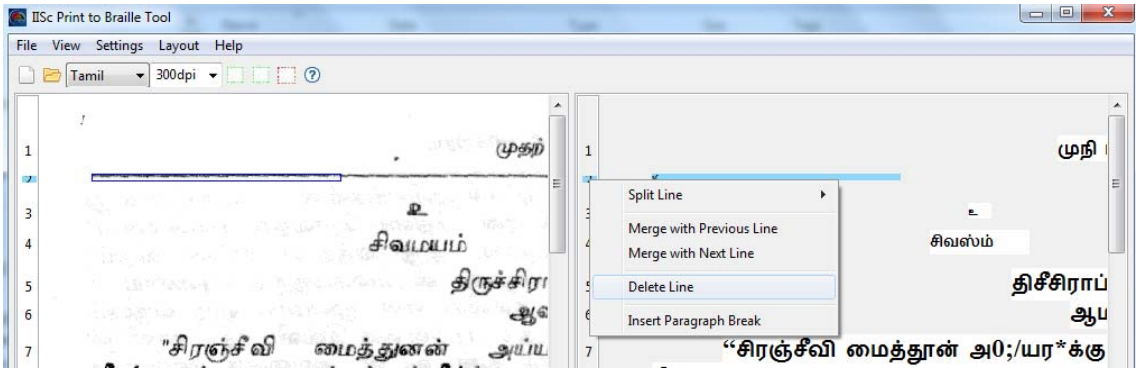

Fig. 5. Facility provided by the tool to correct wrong line segmentation by the OCR.

Standard XML

Figure 6 shows a part of the XML file automatically saved by the tool for a page of text. The text block, text line and word numbers and their coordinates are saved in a hierarchical fashion. This can be used to reconstruct the original document with its logical structure, if required. Also, OCR error analysis can be performed, if the corresponding ground truth is available at any of these levels.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<page xmlns="http://mile.ee.iisc.ernet.in/schemas/ocr_output" skew="0.0" ImageLoc="">
    <block type="Text" BlockNumber="1" rowStart="362" rowEnd="488" colStart="559" colEnd="1833">
        <line LineNumber="1" rowStart="392" rowEnd="460" colStart="559" colEnd="1833">
            <word unicode="4." WordNumber="1" colStart="607" colEnd="0"/>
            <word unicode="இத்தாலியில்" WordNumber="2" colStart="0" colEnd="0"/>
            <word unicode="பாசிசம்" WordNumber="3" colStart="0" colEnd="0"/>
            <word unicode="கி.மி." WordNumber="4" colStart="0" colEnd="0"/>
            <word unicode="1922-கி.பி." WordNumber="5" colStart="0" colEnd="0"/>
            <word unicode="1945" WordNumber="6" colStart="0" colEnd="1801"/>
        </line>
    </block>
    <block type="Text" BlockNumber="2" rowStart="498" rowEnd="2930" colStart="297" colEnd="1153">
        <line LineNumber="2" rowStart="522" rowEnd="573" colStart="297" colEnd="1153">
            <word unicode="பாசிசம்" WordNumber="1" rowStart="522" rowEnd="573" colStart="402" colEnd="557"/>
            <word unicode="என்னும்" WordNumber="2" rowStart="522" rowEnd="573" colStart="600" colEnd="765"/>
            <word unicode="சொல்" WordNumber="3" rowStart="522" rowEnd="573" colStart="808" colEnd="934"/>
            <word unicode="'உ(சிஸ்'" WordNumber="4" rowStart="522" rowEnd="573" colStart="978" colEnd=
            "1142"/>
        </line>
        <line LineNumber="3" rowStart="580" rowEnd="633" colStart="297" colEnd="1153">
            <word unicode="என்னும்" WordNumber="1" rowStart="580" rowEnd="633" colStart="308" colEnd="464"/>
            <word unicode="இலத்தின்" WordNumber="2" rowStart="580" rowEnd="633" colStart="486" colEnd="676"/>
```

Fig. 6. A part of a sample XML file from the tool, illustrating the hierarchical structure and the details of the information recorded.

Current users

Worth Trust Chennai has already converted 172 school and college Tamil printed books using this tool and converted them to Braille books, which are already being used by over a hundred blind students. Sri Ramakrishna Math, Chennai is using our tool to digitize out-of-

print books published by them on the occasion of the 150-th anniversary of Swami Vivekananda. AuroLok Trust, Gujarat is using it to convert Tamil books on philosophy related to Sri. Aurobindo. Parankushachar Institute of Vedic Studies, Bangalore is using it to convert a voluminous translation of *Naalaayira Divya Prapandham*. Samskriti Foundation, Mysore is making use of it for similar objectives. Also, the library of Pondicherry University is using this tool for the benefit of the visually challenged users.

Licensing

Currently, the tool is available together with the Tamil OCR for free use by any non-profit organization for digitizing books for the visually challenged or any other social objectives, after a signed agreement. However, people interested in using it for other purposes can contact for a separate licensing agreement.

Some of the enhancements planned are to provide automated segmentation of text from graphics and layout analysis for multicolumn documents and complex layouts. It is also planned to extend the facility for manual selection of text blocks from rectangular to arbitrary shaped regions employing user-defined polygonal windows. Also on the cards is the facility to automate the manual insertion of information such as “image removed” at locations, where the graphics or photos have been removed. Further, it is intended to make the automatic correction of multiple occurrences of the same recognition error across a book more robust by carrying out sophisticate verifications.

Acknowledgments

The authors thank Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India for funding a national level research consortium project on OCRs in Indic scripts and thus facilitating the development of this socially useful tool. Ours thanks are also due to the undergraduate interns from RVCE and through the summer fellowships of Indian Academy of Science. and MILE lab project assistants for contributing to the development of some of the sub-modules of the PrintToBraille tool and to Worth Trust, Chennai for their useful inputs and feedback.

References

[1] IISc-MILE Print to Braille Tool - an open-source project on code.google.com. http://code.google.com/p/ocr-gui-frontend.
[2] XSD for OCR Output XML. http://ocr-performance-evaluator.googlecode.com/svn/trunk/OCR_Model/src/main/resources/ocr_output.xsd.
[3] JAXB classes for OCR Output XML. http://ocr-performance-evaluator.googlecode.com/svn/trunk/OCR_Model/src/main/java/org/iisc/mile/ocr/model.
[4] IISc MILE TTS Web Demo. http://mile.ee.iisc.ernet.in:8080/tts_demo/.
[5] Open source, downloadable Indic Keyboard Interface, under Apache 2.0 licence. https://code.google.com/p/indic-keyboards/