

# On Integrating Diverse Knowledge Sources in Optical Reading of Devnagari Script

Veena Bansal  
veena@iitk.ernet.in

R. M. K. Sinha  
rmk@iitk.ernet.in

Department of Computer Science and Engineering  
Indian Institute of Technology  
Kanpur 208016 India

## Abstract

*Optical character reading has been a topic of research for several decades. However, human competing performance is still a distant reality. One of the primary reasons of human beings' superior performance is our ability to invoke varying knowledge stores which are relevant to the given situation and integrate them to arrive at meaningful and consistent interpretation. In this paper, we identify the knowledge sources and discuss about their role in Devnagari script recognition.*

*An optical character recognition system segments text zone into text lines, text lines into words, and words into characters. These characters are then recognized.*

*At each stage, there is a possibility of ambiguity. Ambiguities can be resolved using varying knowledge sources at different levels. Many of these knowledge sources are independent of the specific document under consideration. For example, script composition rules, word dictionary and syntax-semantics of natural language. On the other hand, character shapes, font, layout etc are information specific to the document and can be obtained through training. The domain knowledge also forms part of context. Heterogeneous knowledge sources are integrated with the help of blackboard architecture.*

## 1. Introduction

An optical character recognition system usually consists of two main stages, namely segmentation and recognition. There is a possibility of ambiguity at each stage. The text zone extraction process may miss text zones consisting of text lines of un-

usual heights such as footnotes and headers. The line segmentation may be ambiguous due to tilt and overlapping of text lines. Some of the closely written words may not get segmented into individual words. The characters may be fused due to ink spread or may have unwanted breaks, resulting in wrong segmentation. The recognition process may not be able to correctly identify a character due to wrong segmentation or inadequate set of features considered for classification. Inadequate or wrong training may also lead to incorrect recognition of a character.

With additional help from various knowledge sources[8], [6], [5] at each level, many ambiguities are resolved. The text zone is extracted from the image by a preprocessing stage. The extracted text zone is segmented into text lines which have horizontal blank space between them. The height information of extracted text lines is analyzed and the most frequent line height is identified as threshold height. The text lines which have height more than the threshold height are checked for a possible fusion of two text lines. In Devnagari script, every word has a header line. Header lines of all words of a line are at the same vertical position. Consequently, header line is the most dominating hori-

zontal line in a text line. If more than one header line is found in a text line and its height is more than threshold height, the text line is broken into two text lines. The segmentation process segments a word into characters which are vertically separate from their neighbours. The threshold width and height are found out by clustering height and width information of the segmented characters. The characters which are taller than threshold height and wider than threshold width are suspected to be fused characters which might need further segmentation. Further segmentation is not attempted till the the output of recognition process is available. If a suspected fused character is rejected or recognized with low confidence by the recognition process, an attempt is made to re-segment the character. The knowledge of various joining patterns of the script guides the segmentation process.

The recognition process tries to recognize characters and words with the help of prototypes of features acquired by a separate training phase. These prototypes are referred to as external knowledge sources as they are available before the recognition process is invoked. The recognition process is likely to successfully classify the characters for which the prototypes are present. The fused characters are either mapped to another character resulting in low confidence figure or get rejected. Partially recognized word is searched in the word dictionary for candidate words. Characters corresponding to unrecognized characters in selected words provide a set of hypothesis for unrecognized characters. The segmentation process is invoked again to further segment the suspected fused characters which have been rejected or have been recognized with low con-

fidence. The recognition process is invoked again to recognize the segmented characters. The hypothesis generated by dictionary are provided to the recognition process as candidate classes.

The processing is an amalgamation of modules which work in a top-down and/or bottom-up manner. A phase is revisited to verify its results with additional transient knowledge acquired either from a later module, an earlier module or the same module. The knowledge sources are integrated with the help of a blackboard architecture. This architecture facilitates addition of more knowledge sources as they become available. Heterogeneous ways of accessing knowledge sources is also supported. The contribution of each knowledge source is visible to rest of the knowledge sources. In this paper, an attempt has been made to identify knowledge sources for optical reading of Devnagari<sup>1</sup> script. We highlight some of the relevant features of Devnagari script next. Various knowledge sources and their role is also described. Experimentation results are also presented.

## 2. Relevant Features of Devnagari Script

Devnagari script has 40 characters which can be written as individual symbols in a word. The characters may also have a *half* form. A *half* character in most of the cases is required to touch the following character, resulting in a composite character. Some characters of Devnagari script take the next character in their shadow because of their shape. The script has a set of modifier symbols which are used only to modify a character. These symbols are placed either on top, at the bottom, on the left, to

---

<sup>1</sup>Devnagari is the script for Hindi which is official language of India. It is also the script for Sanskrit, Marathi and Nepali languages.

the right or a combination of these. Some examples showing all the above cases are given below.

Top modifiers are placed above the *shirorekha*, which is a horizontal line drawn on the top of the word. The lower modifiers are placed below the character which may or may not touch the characters. More than one lower modifier may also be placed below one character. A character may be in shadow of another character, either due to a lower modifier or due to the shapes of two adjacent characters. A composite character is formed, when a character in half form, is written followed by another character. A half character is intentionally written to touch the following character. A half character followed by a particular character takes a different form as shown below: The above modifier is referred to as *Rkar* modifier. Character of the script have varying height and width. Some examples are shown in figure 1. It is observed that a word can be divided into three stripes: a core strip, a top strip and a bottom strip. Top and bottom strips have only modifiers whereas the core strip has the composite characters. Composite character is a combination of half characters, characters and symbols. A composite character may just be a character as well. The top and bottom strip may be empty for a word, only the top may be present or just the bottom or both. The top strip is separated from the core by a horizontal line, usually referred to as *shirorekha*. The lower strip is below the core strip but no feature of the script separates the two.

Above examples give an idea of the complexity of the script and various ways in which a character takes form of a composite character.

6.5in know2.ps

Figure 1: Knowledge Sources and Their Role in Devanagari Text Reading

### 3. Various Knowledge Sources and Their Role

Figure 2 shows various stages of a document reading system. At each stage, some external knowledge sources are used to process the image and contribute to a solution. The transient knowledge sources are generated internally by the system. Each phase and role of the knowledge sources is explained next.

#### 3.1 Text Zone Extraction

A document may contain text of different sizes, fonts and images. The document is divided into blocks of homogenous connected components. Some basic features of the blocks are used to classify them into text, graph, images etc. Without any knowledge of the document layout, text of unusual height, such as titles, headings or footnotes may not qualify as text lines. However, a two pass mechanism makes an attempt to identify the page layout with the help of document layout knowledge [5]. The text zone extraction process looks for titles, headings, footnotes etc. at expected places based on page layout knowledge. During segmentation process, heights of extracted text lines is saved for the use of other modules.

#### 3.2 Text Line Extraction

Text lines obtained in earlier phase may need further segmentation. Some text lines may be fused due to overlap between the two consecutive text lines. In Devanagari script, the overlap occurs due to lower modifiers of one text line and top modifiers of the following text line. *Shirorekha* is the most dominating horizontal line in a text line. This infor-

mation enables the segmentation process to locate fused text lines. The statistics about line height is analyzed to identify the expected line height. The region between two *shirorekha*, constrained by expected line height, is investigated for splitting.

### 3.3 Segmentation of line into Words

A text line is segmented into words. In Devnagari, *shirorekha* helps in identifying a word as every word has a horizontal line above the core strip. There are few exceptions and *shirorekha* may have a break. It is also observed that for some characters, the break in *shirorekha* and inter-character gap are vertically aligned. This leads to wrong word boundary identification. This ambiguity cannot be identified and corrected at this level. The word level knowledge locates wrong word segmentations and identifies proper word boundaries. For example, in English, an OCR output as *goods chool* which is due to wrong word level segmentation of true input *good school* is not even suspected till word *chool* is rejected by the word verification process.

### 3.4 Segmentation of Words into Characters and Symbols

In last section, we described various ways in which composite characters are formed in Devnagari. These lead to a large number of character fusions (conjuncts) and character overlaps (either due to shape or due to modifiers) which needs to be tackled both at the time of training and recognition. A simple strategy is to treat all the composite character as an atomic unit. But this will lead to a large character set besides being an unnatural way of dealing with Devnagari script as it is a logical composition of its constituent symbols. Therefore, character segmentation process is more involved for Devnagari script compared to Roman

script, even for ideal text. As explained earlier, Devnagari script is written into three strips; top, core and lower strip. Initially, top strip is separated from rest of the word with the help of *shirorekha*. The symbols in the top strip which are vertically separate are extracted. Similarly, the region below *shirorekha* is segmented into vertically separate characters. Some of these characters may be conjuncts or have a lower or rkar modifier. Since the lower modifiers are placed below a character, height of the character with a lower modifier increases. The height of all segmented characters is divided into clusters. Statistical information shows that about 20% of characters have a lower modifiers. The height corresponding to the cluster, which contains maximum number of characters, is stored as threshold height in transient statistics knowledge source.

All characters which are taller than the threshold height are suspected to have a lower modifier. Threshold height also helps in locating the lower modifier region.

Similarly threshold width is also calculated for a text line. All characters which are wider than this threshold width are suspected to be composite characters. The segmentation algorithm for composite characters uses structural properties of the script and joining patterns knowledge.

These features are visually extracted for characters and symbols of the script.

### 3.5 Character Recognition

Three types of knowledge sources are available to the recognition process. The visually extracted features include position of the vertical bar in a character and horizontal placing of the character in the

core strip. These features are independent of the font in addition to being reliable. Structural and statistical prototypes are obtained during an off-line training phase. A number of samples extracted from real environment are used for the training process. A text image and true file are provided to the trainer. The training process tries to establish a correspondence between the extracted symbols and true symbols. The linearization of two dimensional symbols is done as the true symbols are provided as a sequence. The training process performs a preliminary recognition to detect wrong mapping between image and the true symbol [3]. This checking is essential in the presence of fused and broken characters as the true text may not be the same as obtained by the segmentation process. Structural [2], [9] and statistical features both may be extracted and stored. The structural description generator also uses structural properties of the script to generate natural descriptions. Vertical bar of a character is the primary stroke where rest of the strokes form junctions. Implicit use of this knowledge has drastically reduced the number of descriptions generated[9]. Statistical features are sensitive to font style which makes them less suitable for multifold text document. The recognition process may selectively use these features.

A testing phase reveals set of characters which the recognition process tends to confuse. In noisy environment, the output of the character recognition process may be augmented by confusions of the output to ensure the presence of true character in the output. These confusions are used as one of the knowledge sources.

### **3.6 Composition of Symbols and Characters into Valid Words**

A character might have been decomposed by the segmentation module into its constituent symbols and characters for recognition module. Natural breaks due to lifting of pen might have resulted into multiple components of a single character. In both the cases, the constituent symbols of a character must be composed back. This module composes them back according to the set of composition rules of the script [6]. The composition is done both at the symbol level and character level. The composed characters must form syntactically valid Devnagari words. Symbols which don't form legal words are corrected if possible with the help of the composition rules. More than one valid word may be formed as an output of this stage with alternatives provided by character recognition process for some of the symbols/characters of the word. This set of rules is available as a knowledge source to the composition processor.

### **3.7 Word Hypothesis Generation and Verification**

A partitioned word dictionary [7] is available to this module. The words generated by composition processor can be verified. In case of a non-dictionary word, alternatives can be suggested by this module.

A lower level module can also use this module to generate word expectations based on partially recognized word or based on word envelop information. These expectations can guide the recognition process in eliminating some of the classes from candidate classes. An addition may also be made to the candidate classes to cover the word level hypothesis.

Words, which are not present in the dictionary, may contain some substitution errors. The charac-

ters which have been recognized with low confidence may be substitution errors. These errors can be corrected by selecting alternative words from the dictionary in a constrained manner. These constraints may come from the knowledge of the envelop of the input word, character confusion knowledge source or from size statistics. A word dictionary and even a phrase dictionary can be used to verify lower level hypothesis. A word may get mapped to another dictionary word. Such errors can be detected at the sentence level with the help of syntax knowledge. If the mapping has not changed the syntactic class of a word, syntax knowledge will accept the word. The semantic knowledge may be required for such errors. In our present system, we use word level knowledge but no attempt has been made to use syntax or semantic knowledge.

#### 4. Implementation Details and Experimentation

The knowledge sources described in previous section are heterogenous in nature. Consequently, heterogenous ways of accessing them is also to be provided. A knowledge source may contribute to the solution in either top-down or bottom-up manner. For example, the word dictionary may verify hypothesis generated by lower level modules as well as generate hypothesis for lower level modules. Consequently, a sequential control flow is not sufficient. A particular knowledge source may be applied to the same part of the image again after receiving contribution of other knowledge sources. The segmentation process highlighted earlier is reinvoked based on the statistics about the size and the outcome of the recognition process. We found the blackboard architecture [1] is a suitable architecture for integrating varying

the knowledge sources. The main component of the architecture is a blackboard which contains the solution(s). The knowledge sources make their contributions by changing to the solution board. The control mechanism which invokes the knowledge sources is not defined by the architecture.

In our implementation, the knowledge sources are represented as procedures and functions. The control structure is also a procedure which looks at the set of candidate classes for a character and invokes appropriate knowledge source. The segmentation of image into vertically separate characters and symbols is done and transient statistics about their height and width is stored for future use. A segmented image is posted on the blackboard along with the candidate classes which consists of complete Devnagari character set. The recognition process is hybrid in nature. The reliable features are applied as filters to the candidate classes. Other features are used to improve the confidence figure of the candidate classes. The composition process forms syntactically valid words which are verified by the dictionary. The high-confidence non-dictionary words are accepted as proper nouns [4]. The process terminates when a word is verified by word dictionary or the associated confidence reaches a preset threshold.

Figure 3 shows words of a text line in its first column. The second column shows segmented characters and symbols. Below each symbol, the output of the recognition process is shown. Corresponding to some of the characters more than one characters are produced as output from the recognition process. All choices for a character are shown below the image of the character. The composite char-

acters have been segmented and shown below the image of composite character. The recognized characters and symbols are composed back and verified through the dictionary. All the true words were verified. We have tested the system on multiple fonts and obtained upto 96% recognition rate at character level. Most of the errors are due to breaks in the characters. Our system has been designed to handle the composite characters. However, no effort has been made to take care of breaks in characters so far.

## 5. Conclusion

In this paper, we have described relevant knowledge sources for Devnagari text reading. We observe that there is considerable improvement in the segmentation results leading to better overall recognition rate. The segmentation algorithm by [2] fails on highly touching characters such as "oo", "oe" and "od" etc. in Roman. [10] gave another discrimination function to cover highly touching characters. [10] made use of joining patterns in an implicit way. The value of parameters in their segmentation algorithm give a hint on the touching pattern. However, the joining pattern information is not used by the recognition process. The knowledge of various joining patterns and set of characters for each type of joining pattern can help the recognition process in eliminating some of the character classes from further consideration as demonstrated by our implementation for Devnagari script.

## References

- [1] R. Englemore and T. Morgan, editors, "Blackboard Systems", *Addison-Wesley Publishing Company*, 1988.
- [2] S. Kahan, T. Pavlidis and H. S. Baird, "On the Recognition of Printed Characters of any Font and Size", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 1987, pp. 274-287.
- [3] R.M.K. Sinha and Veena Bansal, "On Automating Trainer For Construction of Prototypes for Devanagari Text Recognition" *Technical Report TRCS-95-232*, I.I.T. Kanpur, India
- [4] R.M.K. Sinha and V. Bansal, "On Devanagari Document Processing", *Proceedings, Int. Conf. on Systems, Man and Cybernetics*, Vancouver, Canada, 1995.
- [5] R.G. Casey, D. R. Furgson *Intelligent Forms Processing* IBM System Journal, Vol. 29, No. 3, 1990
- [6] R.M.K.Sinha. *1987 Rule Based Contextual Post-processing for Devnagari Text Recognition*. Pattern Recognition, Vol. 20, No. 5, pp. 475-485.
- [7] R.M.K.Sinha, *On Partitioning a Dictionary for Visual Text Recognition*. Pattern Recognition, Vol 23, No. 5, pp 497-500,1989
- [8] S.N.Srihari. *1983 Integrating diverse knowledge sources in text recognition*. ACM Transaction, Office Information System 1, 68-87.
- [9] Veena Bansal and R.M.K. Sinha, "On Automating Generation of Description and Recognition of Devnagari Script using Strokes" *Technical Report TRCS-96-241*, I.I.T. Kanpur, India
- [10] Su Liang, Shridhar, M. Ahmad, *Segmentation of Touching Characters in Printed Document Recognition*, Pattern Recognition, Vol. 27, No. 6, pp. 825-840, 1994