Foundational Issues of Document Engineering in Indian Scripts and a Case Study in Telugu

Atul Negi, Kavi Narayana Murthy, Chakravarthy Bhagvati
Department of Computer and Information Sciences,
University of Hyderabad, India
{atulcs,knm,chakcs}@uohyd.ernet.in

In this paper we discuss the core issues relating to document processing of Indian language documents. We emphasize the distinction between the entities language, script, and font. Script grammar, *unique to Indian languages, is used to define the concept of characters in Indian scripts. Issues of encoding, processing, rendering and recognition are presented in the context of script grammars. A deep understanding of these concepts facilitates good system architecture and design of text processing systems, OCR systems, web documents etc. We give an example of an OCR system developed for Telugu, a major Indian language. The OCR system is structured with a core OCR engine which recognizes basic shapes and a separate module that composes script level syllables for a character encoding standard such as ISCII or UNICODE.*

1 Introduction

We take, here, a general view of Document Analysis as a domain within the Document Engineering field. The importance of document engineering due to internationalization and the need for local language support in multi-lingual countries like India cannot be underestimated.

Here we state that the success of a document processing system lies in its deep understanding of the foundational structure of the language and script it is designed to serve. We emphasize that Indian languages and scripts are characterized by certain unique aspects, differing significantly from other language families of the world. These issues have not been very well understood in the document engineering discipline. In this paper we sketch the relevant features of Indian languages and scripts and show their implications for document engineering. Text representation and processing, structure of web documents and issues relating to Optical Character Recognition (OCR) are taken up. Examples are included from the *DRISHTI* OCR system developed by us for Telugu and other Indian languages. However, the issues raised here are generally applicable to all Indian languages and scripts.

This research was supported by the University Grants Commission's project entitled "Language Engineering Research" under the UPE scheme, and the Resource Center for Indian Language Technology Solutions from the Ministry of Communications and Information Technology, New Delhi.

2 Indian Language Scripts and Characters

India is a country of one billion people, about one sixth of the world's population. India is also an ancient civilization, dating back at least four thousand years. There are as many as 150 different languages spoken in India today. Of these, only 18 major languages have been given constitutional recognition. These major languages include the official languages of the federal states of India and are among the most widely spoken languages of the world. The scope of this paper is restricted to the major languages and in particular to Telugu, a Dravidian language spoken mainly in the southern state of Andhra Pradesh. Telugu is the second largest spoken language in the country and is also one of the most complex. Our focus here will be on issues relating to the Telugu script.

2.1 Indian Scripts

The difference between language and script is sometimes missed out, especially where there is a one to one correspondence between Languages and Scripts. Language *is* speech and writing is only an artifact. A language can exist even without a script, and there *are* several languages without script. Further, a given language may be written in several different scripts. This is indeed a fact as far as Indian languages are concerned. For example,

Sanskrit is written in not only the Devanagari script but also in almost all other scripts. However Telugu is mostly written in the Telugu script.

2.2 What is a Character?

There are different systems of orthography and their formalization in the form of scripts. English and other European languages are alphabetic in nature - i.e. a small set of alphabets are used to compose words using a system of spelling. Chinese script, on the other hand, is an ideographic script - the script consists of picture elements that signify some meaning.

Indian scripts are said to be mostly *syllabic* in nature - the writing depicts, more or less in a one to one fashion, the various sounds in the languages. The appropriate unit of sound representation chosen is the *syllable*. Thus the word 'Telugu' consists of three syllables 'te', 'lu' and 'gu' and it is exactly these three units that we write when we write the word 'Telugu' in the Telugu script.

There are, however, some differences between the spoken syllable and its written counterpart. Spoken syllables can be of various types - V, CV, CVC, CVCC, etc. where V represents a vowel sound and C, a consonant sound. However, the written units are always of the C*V kind - zero, one or more consonant sounds followed by an vowel sound. These units of writing are called *akshara*'s. It must be noted, therefore, that the correct terminology for the units of writing in Indian scripts is *akshara*, not syllable, although one finds the term syllable also in literature. We will always use the term akshara for the units of writing in this paper. *A Character in an Indian script is thus really an akshara.*

2.3 How Many Characters?

Indian languages generally do not prescribe spelling rules, and all spoken sounds must be directly represented in the orthography. Therefore the number of possible akshara's is naturally very large. Theoretically an unlimited number of consonant sounds may be combined in C*V combination, leading to an infinite number of possible aksharas. Practically we limit the number of consonants in a consonant cluster and the largest consonant cluster known is the 5 consonant cluster in the Sanskrit word 'kaartsnya' involving the consonants 'r', 't', 's', 'n' and 'y' in a single akshara. If all possible C*V combinations upto 5 consonants were allowed, then the number of possible akshara's would still be extremely large - of the order of ten billion! Clearly, not all combinations are possible. Only about 20,000 aksharas are found in the text corpora available in Indian languages. Further, our studies on corpora have shown that about 5000 aksharas account for more than 99% of all the words used in all the major Indian languages.

It is clearly not possible to represent or code so many aksharas directly when writing. Indian scripts have evolved a unique and ingenious method - a *script grammar*, to achieve elegance, simplicity and economy while being complete, in the sense that every possible akshara is represented. This is presented in the next section.

3 A Grammar for Scripts

Akshara's, (or syllables) are composed of consonants and vowels. A small Finite State Machine is sufficient to recognize and generate all valid aksharas and at the same time prohibit ungrammatical combinations. Note that the notion of grammatically valid and invalid combinations does not exist in other systems of writing. A word may be spelled correctly or incorrectly in English. A word may be a valid English word or it may not be. But there is nothing that makes a spelling ungrammatical. Ungrammatical combinations can never occur in the language, not even in proper names or newly coined words. A script level grammar is unique to Indian languages.

All valid aksharas can be constructed from a small set of less than 100 symbols using the script grammar. There are about 40 consonants and about 15 vowels. All aksharas are composed from these basic units. *Aksharas are the atomic units of writing and consonants and vowels are thus sub-atomic units used to construct a grammar for all valid combinations.* The finite state grammar shown in Figure 1 can generate exactly the (infinite) set of valid aksharas:

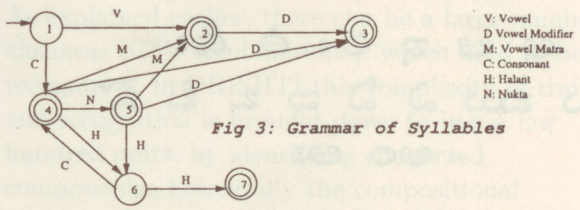

Figure 1: Script Grammar

The following points may be noted with regard to this script grammar. In Indian scripts, the written shapes of vowel sounds when they occur in conjunction with consonants are vastly different from the shapes used to represent pure vowels. In keeping with this significant property, this script grammar employs two sets of symbols, one for pure vowels and the second, called *vowel maatras*, for the vowel sounds in conjunction with preceding consonants. (Consonants may also show variations in shape depending on whether they occur independently or in conjunction with other consonants. The choice is entirely conditioned by the occurrence of a cluster and these variants are not depicted explicitly in the above grammar.) By definition, vowels can be pronounced independently whereas consonants can only be pronounced along with a vowel sound. It is a well established convention in Indian languages, therefore, that consonants are assumed to have an implicit 'a' vowel in them unless this vowel is removed by an explicit symbol called 'halant'. A consonant without an implicit vowel is called a *pure consonant*. Thus consonant clusters are formed by combing pure consonants. When a consonant combines with a vowel maatra, the implicit 'a' vowel is replaced by the corresponding vowel. (Another possible approach could be to view all consonants as pure consonants by default and add the 'a' vowel just as any other vowel as and when required.) Further, vowel modifiers, which are neither pure vowels nor pure consonants, are accommodated as final components of aksharas. Lastly, the grammar allows not only C*V syllables but also vowel-less segments, which are obtained by a sequence of two halants.

Learning to read and write Indian scripts involves recognizing the individual shapes for isolated and conjunct consonants, vowels, vowel maatras and vowel modifiers. Once these basic shapes are mastered, the script grammar dictates the way aksharas are formed. There is no need to memorize any spelling rules. Figures 2-5 depict the basic shapes used in the Telugu script:

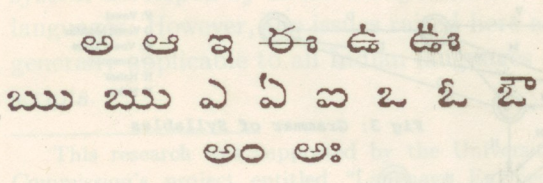

Figure 2: Telugu Vowels

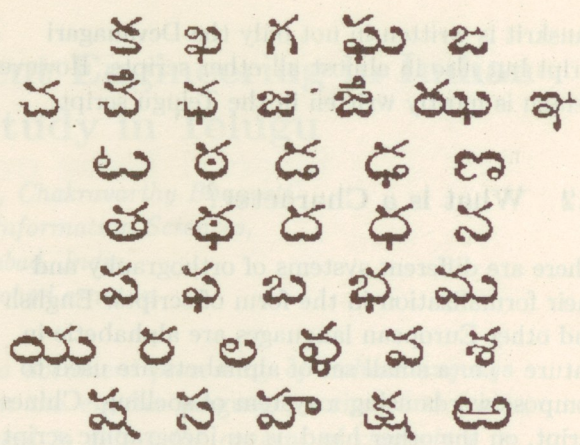

Figure 3: Telugu Consonants

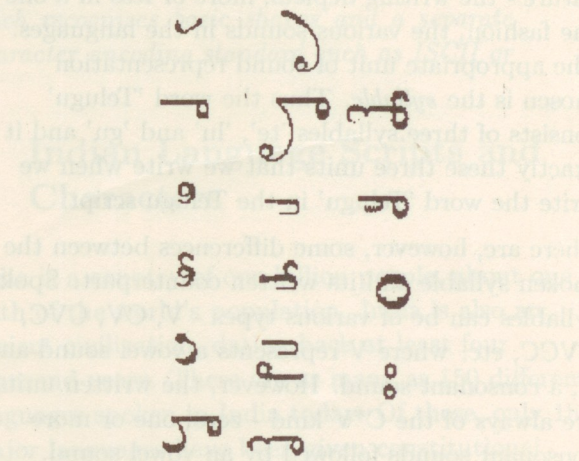

Figure 4: Telugu Vowel Maatras

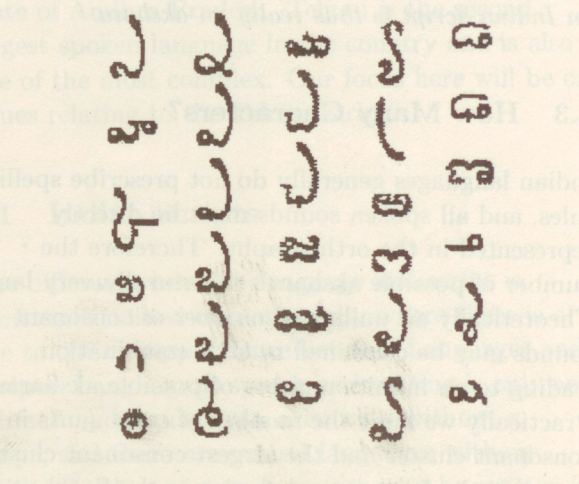

Figure 5: Telugu Secondary Consonants

4 Fonts and Glyphs

The graphical shapes of the aksharas in Indian scripts are quite complex. Aksharas are of variable width and the vowel maatras and conjuncts embellish the base part on the top, on the right side, left side or at the bottom. Text in Indian scripts is not simply a linear sequence of more or less equal with and equal height alphabets. This throws up many challenges in the computer processing of Indian scripts and good design calls for thorough understanding and careful consideration of many issues.

The shapes of the aksharas in Indian scripts are quite complex and aksharas can only be composed from smaller, simpler shapes. These elementary shapes used for rendering are called *glyphs*. Glyphs need to be designed for ease of composition into aksharas. The units of the script grammar are not the best choice for the purpose. Indian language fonts therefore use sets of glyphs that do not have one to one correspondence with the script grammar elements. Several symbols may be combined to form a single glyph and a single symbol may actually be composed of several glyphs. This is shown in Figure 6.

A Font is a set of glyphs placed in a suitable format. Glyphs may be defined as bitmaps or, more often, as second or third degree splines. Fonts for Indian languages ideally must also specify the rules for choice of glyphs and for glyph placement and composition. Further, the rules for mapping from character encoding standards must be specified.

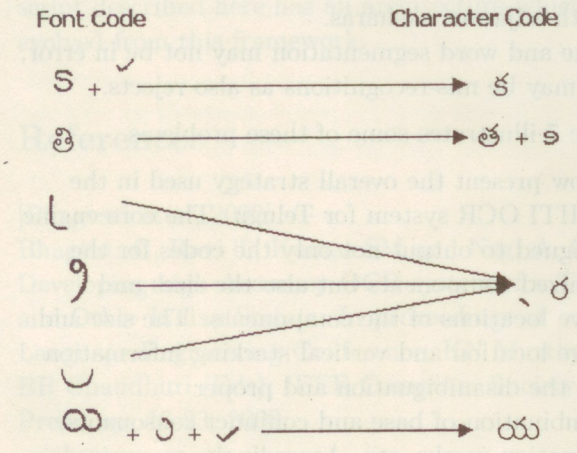

Figure 6: Examples of Font-to-Character Mappings

4.1 Character Encoding Standards: ISCII and UNICODE

We have seen that Indian scripts are phonetic - what we write is what we speak. Orthography is actually a graphical rendering of sounds. In ISCII (Indian Script Code for Information Interchange) the sounds rather than the graphic shapes used in different scripts are encoded to obtain a universal character encoding scheme that captures the essential nature of Indian languages. UNICODE is seen to be a variation on ISCII. The ISCII standard specifies the script grammar and provides an 8 bit code for the vowels, consonants, vowel modifiers and vowel maatras. There are about 40 consonants and about 15 vowels and the total number of symbols to be encoded is less than 128. The final output of an Indian Language OCR system needs to be in an ISCII or UNICODE encoding so that suitable tools for text represenation and processing can be used to edit the recognized text and produce a useful final form. Here due to space constraints we do not talk of the issues related to text representation and processing. Instead we go ahead and describe the architecture of the Telugu OCR Engine.

5 DRISHTI: An OCR Engine

In the previous sections we have built a comprehensive foundation for the different aspects of document image analysis of an Indian Language system. Here onwards we shall concentrate on the Telugu OCR system. Here we describe briefly the architecture of an OCR system for Telugu script called DRISHTI, which highlights the concepts described earlier. DRISHTI as a system was first described in [Negi et al, 2001], and subsequently improved in [Bhagwati et al, 2002]. Presently DRISHTI handles good quality Telugu documents with a single column of text scanned at 300 dpi. As reported [Bhagwati et al, 2002] it has a recognition accuracy at the glyph level of about 97%. It has been tested on several kinds of input ranging from newspapers and popular novels to laser printed text.

As explained earlier, there can be a large number of aksharas (C*V combinations), which need to be recognized. In DRISHTI this complexity of training and recognition is brought down to just a few hundred units, by identifying connected components. Essentially the compositional approach is used where connected components from the binarized image are the units which are

recognized. This bypasses tricky issues of attempting to segment maatras from the base characters. Fortunately secondary consonants in clusters are distinct connected components in Telugu and they need not be dissected from their main consonants like in Devanagari or Bangla scripts. This idea has also been used by other approaches to Telugu OCR [Vasanthalakshmi and Patwardhan, 2002], [Negi et al, 2003].

Line, word and character separation are the stages in isolation of the connected components which are then fed to the recognition engine. Telugu characters are mostly rounded and devoid of straight strokes. Simple projection profiles are not suitable for the complexity of Telugu orthography. A smearing technique using RLSA (Run-Length Smearing Algorithm) [Wong et al, 1982] with both vertical and horizontal thresholds estimated from the document are used. This helps to form words and lines. Connected components are then isolated from the words found by smearing. The layout analysis yields position information.

Recognition of connected components is done using advanced template matching with fringe distances. Templates of a standard size are matched with input scaled to the template size of 32x32 pixels. Fringe distance matches were suitably improved by use of distance techniques for binary template matching as described by Tubbs [Tubbs, 1989]. The complete OCR algorithm is given below:

1. Read in an input binary image
2. Segment the image into lines and words
3. Extract the connected components from each word
4. For each component
   (a) Normalize size to match stored templates
   (b) Compute fringe distance map
   (c) Compute fringe distance from all templates
   (d) Output template with smallest fringe distance
   (e) Convert template code to ISCII code
5. Store ISCII output in a file

Here we should note that the step 4(e) above is not at all a trivial task. This step is perhaps unique to the Indian OCR systems. We describe the details of this process in the following section.

6 From Recognition to Text

The OCR engine outputs codes corresponding to connected components recognized. There as as many codes as there are different connected components. These codes need to be mapped onto a character encoding standard such as ISCII or UNICODE before it can be viewed and edited as text. There are several factors that make this text reconstruction process quite complex.

1. The DRISHTI OCR engine recognizes connected components, which do not necessarily correspond to aksharas or the building blocks of aksharas - vowels, consonants, vowel maatras etc. A connected component may correspond to one building block or fractions or multiples thereof.
2. A code output by the core engine may actually stand for several different things. The shapes of certain consonants are same when used as base characters or as secondary consonants in clusters. The codes output by the OCR engine will be the same in both cases but their character level representations will be different - a preceding halant needs to be inserted when the consonant is a secondary consonant in a cluster.
3. In some cases, the presence of a connected component implies a change in the character recognized without it.
4. The order in which the OCR engine outputs the codes for the connected components does not necessarily correspond to the order in which aksharas are composed. The OCR engine basically works left to right and top to bottom on the scanned image. Parts of one akshara may get mixed up with adjacent aksharas.
5. Line and word segmentation may not be in error, there may be mis-recognitions as also rejects.

Figure 7 illustrates some of these problems.

We now present the overall strategy used in the DRISHTI OCR system for Telugu. The core engine is designed to output not only the codes for the recognized components but also the sizes and relative locations of the components. The size and relative location and vertical stacking information aid in the disambiguation and proper re-combination of base and conjunct consonants, punctuation marks, etc. Accordingly, recognized components are classified into base character, secondary consonant in a consonant cluster, vowel maatra and punctuation. Identification of base characters has been found to be very robust. There are only a few known exceptions, all of which can

be disambiguated through the corresponding character level codes. As a rule, aksharas start with a base character (full vowel or a consonant) and this rule is used to hypothesize syllable boundaries. Once syllable boundaries are determined, the script grammar is used to re-order the other components to build valid aksharas. This overall strategy has been found to be working quite well.

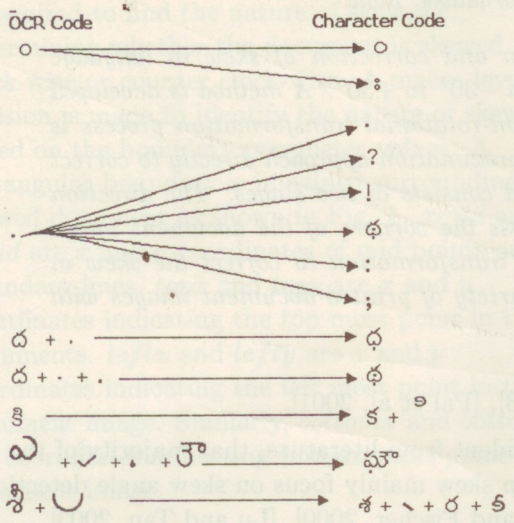

Figure 7: Examples of OCR Output Code to Character Mappings

7 Conclusions

We have described a script grammar which defines a framework for the representation and processing of Indian language documents at the language, script and font levels. The OCR system for Telugu script described here has an architecture which has evolved from this framework.

References

[Bhagvati et al, 2002]
Bhagvati C, Ravi T, Kumar SM and Negi A. On Developing high Accuracy OCR systems for Telugu and Other Indian Scripts. In proceedings of Language Engineering Conference, KN Murthy and BB Chaudhuri (Eds), IEEE Computer Society Press, pp. 18-23, 2002.

[Charan et al, 2004]
Charan GS, Kavi NM and Durga Bhavani S. Text Categorization in Indian Languages, 2004.

[Kavi, 2002]
Kavi NM. Issues in Standardization of Character Encoding Schemes, Technical Report, Department of CIS, University of Hyderabad, 2002.

[Kavi and Hegde, 1999]
Kavi NM and Hegde N. Some Issues Relating to a Common Script for Indian Languages. International Conference on Indian Writing Systems and Nagari Script, Delhi University, Delhi, February 1999.

[Kumar and Kavi, 2004]
Kumar GB and Kavi NM. Script Independent Language Identification in the Indian Context. In Proceedings of iSTRANS 2004 International Conference - Vol 1, RMK Sinha and VN Shukla (eds.), Tata McGraw-Hill Publishing Company Ltd, pp 74–81, 2004.

[Negi et al, 2001]
Negi A, Chakravarthy B and Krishna B. An OCR system for Telugu. In Proc. Sixth International Conf. Document Analysis and Recognition, IEEE Computer Society Press, CA, 2001.

[Negi et al, 2003]
Negi A, Shankar KN and Chereddi C. Localization, Extraction and Recognition of Text in Telugu Document Images. In Proc.of Seventh International Conf. Document Analysis and Recognition, Scotland, IEEE Computer Society Press, CA, pp. 1193-1197, 2003.

[Tubbs, 1989]
Tubbs JD. A Note on Binary Template Matching. Pattern Recogition, Vol. 22, No. 4, pp. 359-365, 1989.

[Vasanthalakshmi and Patvardhan, 2003]
Vasanthalakshmi C and Patvardhan C. A Multi-Font OCR system for printed Telugu Text. In Proceedings of Language Engineering Conference, KN Murthy and BB Chaudhuri (eds.), IEEE Computer Society Press, pp. 7-17, 2003.

[Wong et al, 1982]
Wong KY, Casey RG and Wahl FM. Document Analysis System, IBM Journal of Research and Development, Vol. 26, No. 6, pp. 647-656, 1982.