XML Standard for Indic Online Handwritten Database

Swapnil Belhe
CDAC, Pune
swapnilb@cdac.in

Srinivasa Chakravarthy
IIT Madras
srinivasa.chakravarthy@gmail.com

AG Ramakrishnan
IISc, Bangalore
agrkrish@gmail.com

Abstract

This article proposes an improved XML standard for storing online handwritten data in Indian languages. This standard has evolved over a period of two years, and is currently being used by the Consortium for online handwritten recognition of Indian languages, for annotating about 100,000 handwritten words in each of six Indian languages, namely, Tamil, Kannada, Telugu, Malayalam, Hindi and Bangla. In order that the huge amount of data that is being collected is useable by the future researchers, it is preferable that the data is stored in a format that is unambiguous and easy to read. The uniqueness of this refined standard is that it gives quality labels at different levels to the data, and has provision to annotate all the peculiarities of writing the script of the various Indian languages included in the current consortium project. The current format allows the use of automated and semi-automated annotation tools.

1. Introduction

A database containing 100, 000 words each in six Indian languages has been created by the research partners of the Consortium, "Online Handwriting Recognition System for Indian Languages (OHWR)" funded by Dept of Information Technology, Government of India. This database has been created to train and test the recognition engines in Tamil, Kannada, Telugu, Malayalam, Hindi and Bangla. For easy sharing of this crucial data, an XML standard has been created by the members, after many rounds of discussions and update.

The peculiarities of Indian script demands annotation at additional levels. For example, in Kannada, the second consonant in a consonant cluster is printed or written to the right bottom of the first consonant, while the vowel of the second consonant modifies the shape of the first. This necessitates a separate label called "*ottu*", which refers to the above bottom right symbols. Further, there is another label "stroke group", which refers to a set of strokes that forms the right or bottom auxiliary for a main unit comprising a consonant or consonant-vowel combination. There can be two *ottus* or consonant conjuncts and the defined standard has provision to represent all of these, so that someone can access only such bottom auxiliaries from the database, if necessary. For example, ನ್ರು is a consonant-vowel combination (NRU), where the main symbol is ನ (Na) and the ottu ್ರು occurs at the bottom of the main symbol.

2. Annotation Hierarchy

This XML schema is primarily designed for textual ink documents and does not incorporate general ink documents which may contain figures, mathematical equations, etc. The scope of this representation is limited to a single document written by a single writer and is not meant to span across documents. The schema helps categorize the data into multiple levels viz., Page, Line, Word, Akshara, Stroke group and Stroke levels and encapsulates the data collection and writer information along with the actual handwriting sample. This schema is divided into 4 main parts as shown in Fig. 1.

The *DatasetDef* section provides information about the template used for collecting the handwriting sample, the language used in the template and it traces back to the original data collection template. The templates used for data collection are different across languages. The section also provides brief description of the template along with the Institute, where the template is created with contact information.

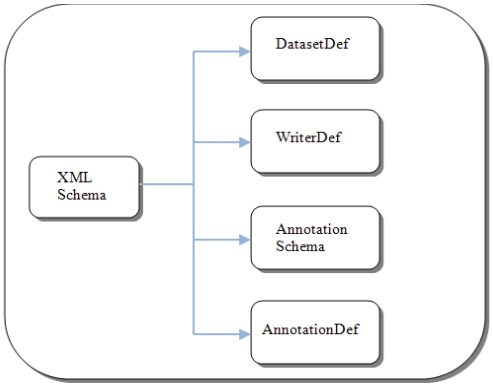

Figure 1: Broad hierarchy of annotation

We performed the preliminary study of sample handwritten data across different languages of India and the variation of writing styles. To enable further study of this trend by different institutes, we have also incorporated extensive writer details in *WriterDef*. The *WriterDef* section provides details about the writer, i.e, the person actually providing the handwriting sample. The details such as age, gender, education level, region, frequency of writing, right-left handedness of the writer etc. are all documented. All the writers selected were native writers in their languages and write at least one page/day. In India, it is broadly observed that the writing style varies across regions, age group and the education level. This XML representation also incorporates different ink devices available and stores their information in *deviceType*.

The handwritten data is categorized into a tree structure. Each document has a number of pages written by the same writer. Each page is made up of a number of lines, words, aksharas, etc. Figure 2 lists the various annotation levels. The *AnnotationSchema* element enumerates the different data elements making up the handwriting data. The *AnnotionSchema* element uses a special element type, the *annotationType*. The *annotationType* gives a list of all possible data element types which could make up the handwriting data. If a new encapsulation level is to be added in the future, say paragraph, it should be added to the *annoType* and then called everywhere else.

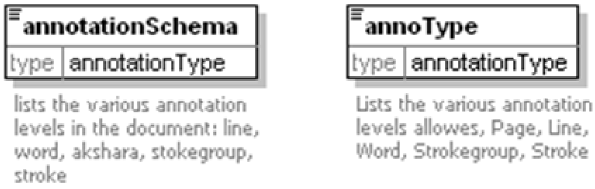
annotationSchema
type annotationType
lists the various annotation levels in the document: line, word, akshara, stokegroup, stroke
annoType
type annotationType
Lists the various annotation levels allowes, Page, Line, Word, Strokegroup, Stroke

Figure 2: Annotation Scheme

AnnotationDef element encapsulates the handwriting data captured by the pen device.

- Each level of the *AnnotationDef* contains one or more elements from the next lower level. There are 2 exceptions to this:
  - Word data sometimes contains stroke data directly. In case of cursive languages like Bengali, the word data cannot be segmented into aksharas; so the next possible level after word is stroke. In the case of languages with a distinct shirorekha (a dominant line over characters), the shirorekha cuts across the aksharas. So, it is attached to the word directly.
  - Akshara data sometimes bypasses StrokeGroup element and contains the stroke data directly. Not all handwriting recognition engines use the concept of StrokeGroups. Attaching the stroke element to the akshara directly takes care of such classification.
- Each level maintains a count of the number of elements in the next lower level. The document element is the highest level of annotation definition; it contains a count of the number of pages in the dataset. A document is a collection of pages written by a single writer.
- Each level also has a number, which is basically a running count of the element type in that group.

3. XML elements used

Document element: Since the document is the highest level and there is only one document for every xml file, it does not have a document number. It is a simple collection of pages.

Page element: Page is a collection of text lines.

Line element: The line element is a collection of word elements. It contains

- the annotation quality, i.e. the *truthLevel* element. Before any annotation is carried out, the *truthLevel* is none; after manual/automatic annotation, the *truthLevel* is labeled. Finally, after the annotation is verified, the *truthLevel* becomes truthed.
- *word count*, i.e., the number of words in the line.

Word element: The word is a collection of aksharas, stroke groups and strokes. It contains many sub-elements as shown in Fig. 5.

- Annotation quality i.e., the *truthLevel* quality.
- *labelDesc* is a complex element with 3 parts;

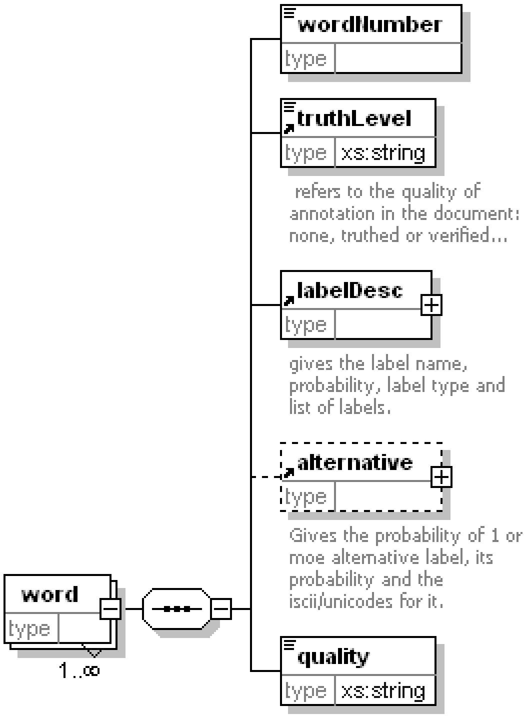
wordNumber
type
truthLevel
type xs:string
refers to the quality of annotation in the document: none, truthed or verified...
labelDesc
type
gives the label name, probability, label type and list of labels.
alternative
type
Gives the probability of 1 or moe alternative label, its probability and the iscii/unicodes for it.
word
type
1..∞
quality
type xs:string

Figure 3: Word element

labelDesc contains 3 parts: in the first, part it contains the elements for human readable ITRANS description. The second part gives the probability that the label is correct. Third part other is a computer readable format encapsulated in the annotationDetails element.

The annotationDetails (see Fig. 4) describes the codeType i.e. ISCII/UTF8/U16, noOfCodes (number of code points required to represent the word) and the actual codeSequence, with each code value separated by a space. For example, for the word aMkana, its code ISCII (Indian Standard Code for information Interchange) would be 164 162 174 164 182 164.

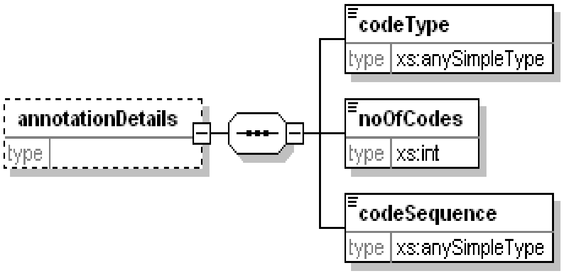

Figure 4: Details of annotation text

Alternative labels are used, if the user wants to provide one or more alternative labels for the data. It has one or more labelDesc as a child element.

As shown in Fig. 5, the word element also incorporates elements for the following:

Class of writing i.e., quality. This describes the quality of the text. It is explained in detail in the next section.

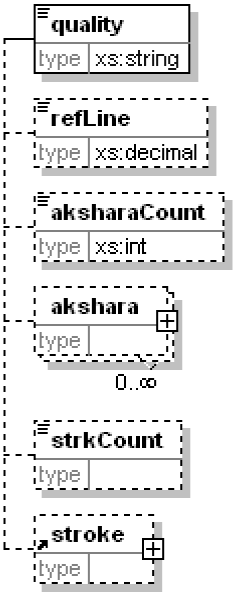

Figure 5: Sub-levels of the word element

A *refLine* is the position of the reference line (shirorekha/baseline), if present and relevant. This gives the ratio of height of text above and below the shirorekha in the case of Hindi and baseline in the case of Kannada or Telugu.

aksharaCount is the number of aksharas in the word.

strkCount is the number of strokes directly under this element.

Akshara element: It encapsulates the data at the level of akshara. An akshara is equivalent to a syllable with all possible consonant-vowel combinations. Akshara element also keeps the stroke group count. The other element_tags within the levels of stroke, stroke group and akshara are the same as that of word level.

Stroke-Group element: The stroke group element is a collection of strokes. It is the collection of any strokes and is not represented as range of strokes. Its sub-elements contains a label Description, Stroke count, one or more strokes. Each akshara may contain more than one stroke group.

Stroke element: The stroke element is the smallest level of encapsulation. This has the actual handwriting trace obtained from the input ink capturing device.
It also contains,
- *truthLevel* for the stroke
- *labelDesc* at stroke level. it does not have annotation details.
- alternative stroke descriptions if any
- a handwriting stroke encapsulated in the *hwTrace* element.

hwTrace *element:* This is a part of the stroke element. It has,

- *Dimension* i.e the number of columns in the trace; Usually we just record the x,y axis data; so the dimension is 2. The pressure information can be easily incorporated by increasing the dimensions. Our current devices only give x,y points.
- The number of points, i.e., *noOfPoints* in the trace
- The actual trace element. This is a list of numbers separated by a space. The number count is a product of the number of points and the dimension element.

Our current representation of digital ink does not record the timestamp information but more general form of this representation will incorporate the same.

4. Quality labels

In this project, there are many languages for which the recognition engines are developed by many institutes with varied performance. Also there is a vast variation in writing styles. Hence, in order to correctly judge the recognition accuracy across centers we formulated a common set of the quality flags which broadly classifies a given word into distinct classes. The implementation of these classes is the same for all languages.

The quality of the word data is tagged at four levels: class A, B, C and D. Class A is data, where each compound character (*akshara*) in the word is segmentable with an automated segmentation logic and further, each of them has been written with the right number and direction of strokes. If the strokes of adjacent *aksharas* overlap a little, and the quality otherwise is good, then the written word is annotated as Class B data. If correct strokes overlap more than 10%, or if two separate strokes are written as a combined stroke, it is labeled as Class C data. This also includes where strokes have been split, resulting in more number of strokes than normally expected. In either case, the akshara must be clearly identifiable by a human being. If there are extraneous strokes or overwriting, and we believe there is a good chance of correction recognition if the extraneous strokes or overwriting is removed in some way, then we term it as Class D data. If the character is unrecognizable by a human or if the strokes have been written in a completely wrong or reverse direction, the data is labled as ‘Reject or R’ class. Each of the attributes, such as, whether the data is human readable, valid Unicode, syllable segmentable and stroke supported, are saved as flags with each word.

5. Acknowledgment

The authors thank Technology Development for Indian Languages (TDIL) Programme, Department of Information Technology, Government of India for funding this Consortium project, and our other partners for their inputs.

6. References

[1] International Unipen Foundation. The Unipen Project. http://www.unipen.org/home.html, 1994.

[2] UPX- The best from UNIPEN and inkML, http://unipen.nici.kun.nl/upx/, 2002.

[3] Mudit Agrawal, Kalika Bali, Sriganesh Madhvanath and Louis Vuurpijl. UPX, “A New XML Representation for Annotated Datasets of Online Handwriting Data”, Proc. International Conf. Document Analysis and Recognition (ICDAR), 2005, pp.1520.

[4] W3C-Multimodal Interaction Working Group. Ink Markup Language (inkML), http://www.w3.org/TR/InkML/, 2006.