Representation and Annotation of Online Handwritten Data

Ajay S. Bhaskarabhatla, Sriganesh Madhvanath
Hewlett-Packard Labs, Bangalore, India
{ajay.b, srig}@hp.com

M. N. S. S. K. Pavan Kumar, A. Balasubramanian, C. V. Jawahar
International Institute of Information Technology, Hyderabad, India
{pavan, balu}@students.iiit.net, jawahar@iiit.net

Abstract

Annotated datasets of handwriting are a prerequisite for the design and training of handwriting recognition algorithms. In this paper, we briefly describe an XML representation for annotation of online handwriting data that uses the emerging Digital Ink Markup Language (InkML) standard from W3C for the representation of handwriting data. We then describe a tool based on the proposed representation that can be used for annotation of digital ink. Ease and speed of annotation are emphasized in the design of the tool. Together, the representation and the tool attempt to address the requirements of creation of annotated datasets of handwritten data in different scripts around the worldwide.

1. Introduction

Annotated datasets of handwriting covering a variety of writing styles are essential for the development and evaluation of modern data-driven handwriting recognition engines. This issue was first addressed in the context of the online handwriting recognition problem by the UNIPEN consortium in the early 1990's [6, 5]. The UNIPEN representation employed ASCII flat files to store handwriting data and associated annotation. The focus of the UNIPEN effort was the recognition of cursive English, and the members of the consortium collected and annotated large amounts of handwriting data in the UNIPEN format. More recently, there have been attempts at creating datasets using the same standard in other languages such as Japanese (Kanji) and Arabic [8].

While research in online handwriting recognition in the context of Roman and many Oriental scripts has continued unbroken for over three decades and resulted in several commercial engines, the same cannot be said for the majority of the world's scripts especially in developing countries. The lack of significant and easily available linguistic resources in the form of annotated datasets of handwriting has been one of the obstacles to research in these scripts. It is clear that many of these resources need to be created, and the creation of such handwriting databases in different scripts calls for a standard representation that is independent of script and allows semantic interpretation of the writing at various user-defined logical levels. The representation should capture information about script, writing style, quality of writing and truth. It should also capture information about writers and the data capture environment. It should support automatic generation of annotation using recognizers, and subsequent manual validation processes. It should keep handwriting data separate from its semantic interpretations and it should support planned as well as casual data collection.

In this paper, we describe *hwDataset*, an XML representation for the annotation of handwriting data that is inspired by the UNIPEN standard, and addresses these requirements. XML is a natural choice for the representation of annotation because of its hierarchical nature and extensibility [1, 4]. The *hwDataset* representation makes use of an underlying XML representation of the raw handwriting data called Digital Ink Markup Language (InkML), a standard being developed by the W3C for the description of digital ink [10]. InkML markup is designed to support the input, storage and processing of handwriting, gestures, sketches, music and other notational languages in Web-based applications, independent of platform. InkML also provides a common format for the exchange of ink data between components such as handwriting and gesture recognizers, signature verifiers, and other ink-aware modules.

InkML provides means for application-specific extensions. By virtue of being an XML-based language, it allows users to easily add specific information to ink files to suit the needs of the application at hand. In this sense, *hwDataset*

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

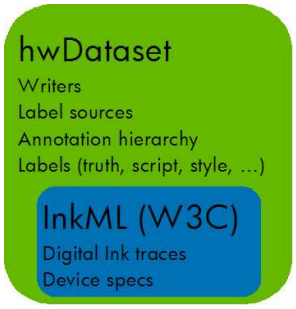

Figure 1. Conceptual relationship between hwDataset and InkML

may be thought of as an application-specific extension of InkML (Figure 1).

In addition to a standard representation, the creation of annotated datasets of handwriting requires effective tools for data collection and annotation. Annotation of ground-truth is a particularly critical and expensive operation that requires careful attention to tool design. Among other things, issues related to non-Roman scripts have to be addressed in detail [9]. In this paper, we also describe a tool for annotation of online handwriting data that can be used to create complete *hwDataset* documents.

1.1 Creation of Annotated Datasets

The creation of annotated datasets of handwriting is generally comprised of sequential data collection and annotation phases. In the data collection phase, handwriting samples are collected using appropriate devices and tools (see for example [2]) and the digital ink captured in files in an appropriate directory structure, using a convenient ink format such as InkML or UNIPEN. This phase is often distributed over different times, places and even organizations. The data collection may be designed or casual. In the former instance, writers with specific skills are recruited for contributing handwriting samples, and asked to write specific characters, words or sentences or a combination, or given other specific tasks. In the latter, digital ink is a by-product of an ink application such as handwritten email or note-taking.

In addition to the handwriting data, metadata pertaining to the design of the dataset and writer profiles is typically, but not necessarily, captured as part of this phase. Basic grouping of the captured ink into the top-level annotation categories corresponding to writing tasks may also be captured in this phase. For example, if the design of data collection requires each writer to write a list of words in different fields of an electronic form, the ink from each field can be automatically grouped into "words" and word-level ground-truth provided. In our own data collection efforts [2], metadata captured during this phase is represented directly as hwDataset documents, with references to digital ink captured separately as InkML documents.

The files of digital ink and any metadata captured as part of the data collection phase form the input to the annotation phase. In this phase, metadata captured as part of the data collection phase can be validated and completed. However the chief activity in this phase is the tagging of ink with labels corresponding to ground truth, writing style, and so on, at different levels of an appropriate hierarchy of annotation levels. Even with the availability of tools, this activity is labor-intensive and several passes may be needed to obtain the desired level of accuracy of annotation. In the general case, annotation may be added across multiple sessions distributed over time and space; not all levels or types of annotation may be provided in one session (or ever !); and multiple annotators - humans or machine algorithms - may provide annotation. For pragmatic reasons, it is generally necessary to have access to partially annotated dataset even as it evolves.

The output of the annotation phase is a collection of hwDataset documents organized into an appropriate directory structure. Each hwDataset document is paired with an InkML document containing the digital ink data referred to in the document.

2. Representation for Annotated Handwriting Datasets

The proposed *hwDataset* representation includes a set of XML elements for detailed annotation of handwriting. The *hwDataset* element is the root of the document and captures metadata about the dataset as part of the *datasetInfo* element, various definitions as part of *datasetDefs*, and hierarchical annotation of handwritten data as part of one or more *hwData* elements. These elements are described briefly in the following paragraphs.

datasetInfo The *datasetInfo* element (Figure 2) captures metadata related to the dataset as a whole. It contains the following elements: (a) *name* - name for referring to the dataset, (b) *category* - type of dataset captured using UNIPEN-style codes, (c) *version* - version number and/or datestamp of dataset publication, (d) *contact* - contact info for dataset-related queries, (e) *source* - the source of collected data, (f) *setup* - physical conditions in which data was collected, and (g) *dataInfo* - information about the data.

The *dataInfo* element in turn contains the following sub-elements: (a) *script* - language/script captured in dataset, (b) *quality* - overall assessment of quality of handwritten data captured in dataset, (c) *truth* - ground-truth of what is

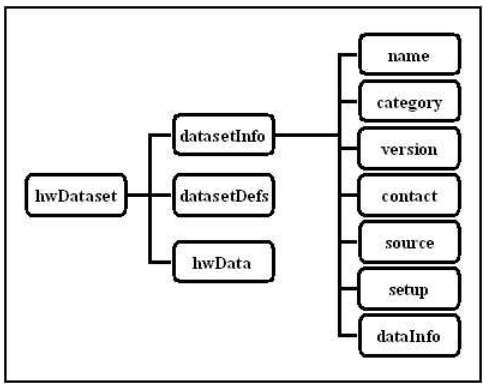

Figure 2. The *datasetInfo* element captures metadata about the dataset

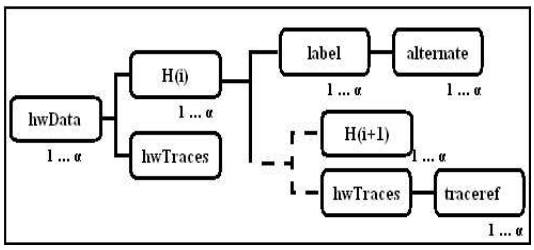

Figure 3. *hwData* element hierarchical organization of annotation

captured such as a word-list, (d) *methodology* - design of data and collection procedure, and (e) *annotationScheme* - description of annotation scheme.

annotationScheme specifies the user-defined hierarchy of annotation by means of a series of *annotationLevel* elements.

datasetDefs The *datasetDefs* element captures information about different writers and sources of labels (annotation) represented in the dataset, and provides the means for referring to them later in the document. It contains the following elements:

- *writerDefs* - declarations of writers as a sequence of *writer* elements
- *labelSrcDefs* - declarations of sources of annotation (human or machine) as a sequence of *labelSrc* elements

Each *writer* element in turn contains a *date* subelement which provides a coarse indication of the time when writing occurred (as opposed to the much more precise trace timestamps in InkML), and a *personal* subelement that captures personal information such as *hand* - left/right handedness, *gender* - gender, *age* - age at the time of capture, *skillScript* - level of skill with script, *skillDevice* - level of familiarity with writing device, *style* - predominant writing style, and *region* - native region.

Each *labelSrc* element contains the following elements: (a) *name* - name of the human/automated source of labels, (b) *description* - description of label source including responsible organization, (c) *time* - approximate date and time of annotation of dataset, and (d) *contact* - contact details of label source.

In addition, an attribute *labelTypes* describes the categories of labels (e.g. truth, quality, script, style) generated by the given source and their character encoding (e.g. UNICODE).

hwData The *hwDataset* document contains one or more *hwData* elements corresponding to different trials or different fields of writing captured from a writer in a single trial. These instances may be distinguished using the *Id* atttribute. Each *hwData* element supports the hierarchical organization of annotation of the corresponding digital ink (Figure 3. It refers to an appropriate level $H(i)$ of the of the annotation hierarchy defined by the user as part of the specification of the *annotationScheme* element. Each level $H(i)$ of the hierarchy contains one or more label elements that captures annotation information at that level. $H(i)$ may in turn contain either one or more $H(i+1)$ elements, or *hwTraces*, the leaf element of the hierarchy that refers to digital ink traces represented using InkML (Figure 3(b)). It is noteworthy that (i) *hwTraces* and the $H(i)$ elements are derived from the *traceRef* element from InkML, and (ii) semantic interpretations of the $H(i)$ levels such as WORD or CHAR are specified as part of the *annotationScheme*.

The *label* element can be used to capture alternative choices of label with confidence values if any, and the exact time of annotation. Although primarily intended to describe the truth value of a particular set of ink traces, it may also be used for describing other characteristics such as writing style, quality and script. The timestamp can be used to generate the history of annotation of a particular unit of writing, spanning different label sources. The alternates can be used to facilitate the process of manual validation of annotation by prompting options for human validation.

Formally, the attributes of label are (a) *id* - identification of label, (b) *labelSrcRef* - a reference to a label source defined earlier, (c) *category* - category of label (e.g. truth, quality, script, style), and (d) *timestamp* - time of the act of annotation.

The *hwDataset* representation attempts to satisfy the requirements for a good representation scheme that were laid out in the introductory section. Script-independence is achieved by supporting different encoding standards for the

truth values, and use of an XML based representation. The representation supports semantic interpretation of the writing at various user-defined logical levels and captures information about script, style, quality at these levels. In addition, these attributes may also be associated with the dataset as a whole, or with specific writers. The *label* and *labelSrc* elements provide the means to capture recognition alternates along with confidences, thus supporting the use of multiple sources of annotation for the automatic generation of annotation and subsequent manual validation. Attributes have closed sets of values whereever possible. Multiple *hwData* blocks can be used to capture different trials or multiple fields of writing within the same trial.

3 Annotation Tool

In this section, we describe a tool that has been designed to facilitate the annotation of online handwriting data using the proposed hwDataset format. While the tool is designed to read and write hwDataset documents, it is also capable of importing pure digital ink in input formats such as InkML, UNIPEN, and simple ASCII encodings of trace data. The tool supports input and output, viewing, editing and annotation of digital ink at different levels of a user-defined hierarchies. The tool is supplemented by a library of basic functions that can be used to access and extract handwriting data from hwDataset documents based on user-defined criteria.

3.1 Software Architecture

The tool implements an open and extensible architecture using plug-ins for different operations such as page segmentation and word recognition. Segmentation plug-ins are implemented for common hierarchical levels such as strokes, words, and lines. The tool also allows multiple plug-ins for the same operation (for example, line segmentation) and selection of a specific plug-in at the beginning of the annotation session. This allows for customization and dynamic selection of these modules. In addition, word recognition plug-ins may be used to partially or fully automate the generation of ground-truth for handwriting data. The tool also provides a basic set of stroke signal processing routines such as mean filter, median filter, one-dimensional Fourier transform, and so on. While not central to annotation, these allow the visualization of the effect of different kinds of signal processing on digital ink. Since all the plug-ins for a given class of operations return results in standard formats, they are handled within the tool in a consistent manner. Sample plug-ins are provided along with the tool, and new plug-ins may be written in C++.

The tool has been developed in C++, using Qt as the GUI toolkit. Currently the tool is supported on two popular desktop operating systems - Linux and 32-bit Windows. Since C++ and Qt are available on many other platforms, the tool can be extended to those platforms.

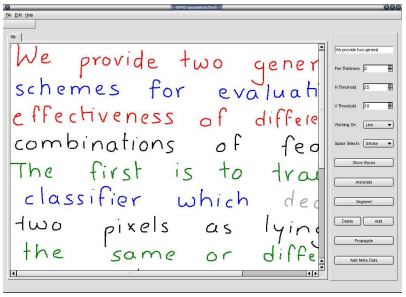

Figure 4. English text being annotated at the line level following automatic line segmentation

3.2 Usage

The user interface of the annotation tool features a workspace for manipulation of digital ink and a tool bar with buttons for all the key operations (Figure 4). Capture of metadata related to the *datasetInfo* and *datasetDefs* elements in the representation is supported through additional screens. Note that any metadata present in the input *hwDataset* document is automatically read and made available for modification through these screens.

Once metadata has been provided, the next important step is specification of the annotation hierarchy. Here the user has a choice of predefined levels such as WORD and CHARACTER as well as the flexibility to define new ones. In general, annotation proceeds by segmenting a digital ink document at a desired level of granularity such as lines or words using an available plug-in, or into individual strokes. The tool currently supports automatic segmentation at the stroke, word and line levels, based on inter-stroke distance patterns. As mentioned earlier, these algorithms are embedded in plug-ins and can be modified by the user.

The segments (stroke groups) obtained as a result of segmentation at a particular level are displayed in different colors using a coloring scheme. These segments generally need to be manually corrected, and the interface supports the use of both mouse clicks and keyboard shortcuts for adjustment of segments and for selection of one or more segments for further annotation.

Once the segmentation has been verified, the user can manually annotate segments by directly keying in ground-truth using a QWERTY keyboard. A single selection-keystroke may be used to end the annotation for the current segment and advance to the next segment. In this manner, all segments at a specific annotation level can be truthed

fairly easily and quickly with minimal overhead. Ground-truth is recorded as an uninterpreted string of characters and hence may be provided using various standard encodings such as ASCII, ISCII and ITRANS, or other custom encoding schemes.

At present, each level of annotation is meant to be provided independently from the others. There are clearly top-down (alignment) and bottom-up (concatenation-based) strategies for generating ground-truth for specific levels from the truth or transcriptions at other levels. These can be modeled as independent label sources of machine type in the current scheme, and may be implemented as plug-ins in future versions of the tool. In addition, annotation at the word level can be partially automated using word recognition plug-ins, as described earlier.

Propagation of annotation is supported by the tool for semi-automated annotation of multiple writing trials by the same writer collected as part of a designed data collection effort. Propagation works by using the entire subtree of annotation from the "source" word and applying it to the target words. The association of character-level annotation to the strokes of a target sample may be performed based on stroke indices, or more intelligently by matching stroke shapes. The tool provides both implementations to support such propagation.

While we have described annotation in the context of ground-truth, the same principles are applicable to other kinds of annotation pertaining to the script, style and quality of writing.

3.3 Efficiency

We have made some preliminary attempts at benchmarking the efficiency of manual annotation of ground-truth using the tool. The efficiency of annotation was measured as the average number of key strokes used for annotating various documents at various hierarchical levels. Handwritten pages containing 200 characters and 50 words on average were taken for the analysis. Character level annotation was found to require approximately 450 key strokes per page on average, and word level annotation around 250 key strokes. The time taken to annotate a page of 50 words at the word level was around 2 minutes for a person with typing speed of 40-50 words per minute.

3.4 Access Library

We have attempted to address the requirements of extraction of information from annotated datasets by providing a library of functions implemented in C++.

The access library provides interfaces to the user to access specific subsets of the information contained in the collection of hwDataset documents based on constraints on element and attributes of the hwDataset document.

(a) (b)

Figure 5. Samples of writing from (a) Telugu and (b) Amharic

3.5 Discussion

Since one of the major motivations behind this effort is the creation of linguistic resources for scripts which are lacking in them, support for non-Roman scripts is an extremely important issue. The ten Indic scripts supporting India's eighteen official languages are good examples of such scripts. These scripts are defined as syllabic alphabets, and the syllabic units of writing are characterized by a large number of strokes and high structural complexity. These syllabic units may be broken down further into simpler shapes for recognition in a few different ways. The representation and tool therefore need to support custom hierarchies, at least below the level of the syllabic units.

Similarly, the encoding scheme for ground-truth (text) can be different for various scripts. For Indic scripts there are a number of competing encoding formats including the 8-bit ISCII, the Indian Script Code for Information Interchange [7], and the 16-bit UNICODE standard. The representation and annotation tool therefore support multiple encoding schemes. However, correct rendering of ground-truth labels in specific encodings often poses a problem.

We have experimented with ASCII, UNICODE and ISCII encodings in limited tests of the annotation tool (approximately 30000 words of English, Hindi and Telugu and 1000 words of Amharic). For Indic scripts, we have used an encoding called ITRANS which captures a transliteration of the text in ASCII [3]. The transliteration is human readable and may be mapped unambigously to ISCII or UNICODE. Samples of some of the scripts used for testing are shown in Figure 5. The tool has been tested using digital ink data collected from different devices such as IBM crosspad, HP TabletPC, HP iPAQ (PocketPC) and external tablets.

4 Conclusions

In this paper, we have proposed an XML-based representation called *hwDataset* for hierarchical annotation of online handwriting data to support handwriting recognition, especially for scripts of the developing world for which very little recognition technology exists. Besides annotaion of ground-truth, the representation supports annotation of other aspects of handwriting such as writing style, quality and script, and accommodates multiple writers and annotation sources of digital ink. The representation builds upon Digital Ink Markup Language (InkML), a draft specification of digital ink being developed by W3C.

We have also described a tool for the annotation of online handwriting data based on the hwDataset representation. The design of the tool emphasizes ease of use and speed of annotation. A plug-in architecture allows integration of custom plug-ins for partial automation of operations such as document segmentation, and generation of ground-truth. The tool, like the representation, is designed to support and has been tested with a variety of scripts.

The *hwDataset* representation and annotation tool are a work in progress, and need to be validated in the context of a wide variety of scripts and writing styles. We intend to make a beginning in this direction by deploying the tool for data collection efforts in various Indic scripts over the coming year. [1]We invite comments, suggestions and participation in the larger effort to create a truly general and useful representation and tools, from those engaged in handwriting recognition research and the creation of linguistic resources, especially for new scripts.

Acknowledgments

The authors would like to gratefully acknowledge inputs from Prof. Mark Liberman of Linguistic Data Consortium, U Penn, Prof Louis Vuurpijl of the UNIPEN consortium, NICI, members of the ink subgroup of the W3C Working Group on Multi-modal interaction and the reviewers of this manuscript.

References

[1] A. P. Lenaghan, R. R. Malyan. XPEN: An XML Based Format for Distributed Online Handwriting Recognition . *International Conference on Document Analysis and Recognition*, pages 1270–1274, 2003.

[2] A. S. Bhaskarabhatla and S. Madhvanath. Experiences in Collection of Handwriting Data for Online Handwriting Recognition in Indic Scripts. *4th International Conference on Language Resources and Evaluation*, 6:2223–2226, 2004.

[3] A. Chopde. ITRANS3.5 - A Package for Printing Text in Indian Languages using English-encoded Input. *http://aczone.com/itrans*.

[4] K. Franke, L. Schomaker, C. Veenhuis, L. Vuurpijl, M. van Erp, and I. Guyon. WANDA: A Common Ground for Forensic Handwriting Examination and Writer Identification. *ENFHEX news - Bulletin of the European Network of Forensic Handwriting Experts*, (1/04):23–47, 2004.

[5] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman and S. Janet. UNIPEN Project of Online Data Exchange and Recogniser Benchmarks. *Proceedings of the 12th International Conference on Pattern Recognition*, pages 29–33, 1994.

[6] International Unipen Foundation. The unipen project. *http://www.unipen.org*, 1994.

[7] Ministry of Information Technology, Govt. of India. Indian Script Code for Information Interchange. *http://tdil.mit.gov.in/standards.htm*.

[8] M. Nakagawa and K. Matsumoto. Collection of On-line Handwritten Japanese Character Pattern Databases and their Analysis. *International Journal on Document Analysis and Recognition*, 7(1), 2004.

[9] M. Pavan Kumar, S. S. Ravikiran, Abhishek Nayani, C. V. Jawahar, and P. Narayanan. Tools for Developing OCRs for Indian Scripts. *DIAR'03 in connection with International Conference on Computer Vision and Pattern Recognition*, 2003.

[10] W3C Multi-modal Interaction Working Group. Ink markup language (inkml). *http://www.w3.org/2002/mmi/ink*, 2003.

[1]Correspondence may be directed to Sriganesh Madhvanath (srig@hp.com)