White Paper

Report ID: 2881817

Application Number: PR-50178-13

Project Director: Peter M. Scharf

Institution: Sanskrit Library

Reporting Period: 6/1/2013-7/31/2016

Report Due: 10/31/2016

Date Submitted: 10/30/2016

Final Report
PR-50178-13
Developing automated text-image alignment to enhance access to heritage manuscript images

Peter M. Scharf
The Sanskrit Library

30 October 2016

1 Introduction

The stated aim of the project was to enhance access to primary cultural heritage materials of India by developing human-validated automated text-image alignment techniques in order to provide access to digital images via related machine-readable texts, lexical resources, linguistic software, and a sophisticated search interface. In particular the project developed software to facilitate text-image alignment of images of Sanskrit manuscripts written in Devanagari script with digital editions of the same works. Since OCR technology by itself is not capable of generating accurate Sanskrit text from an image of an Indic manuscript, or even of a printed book, known characteristics of the manuscript, of the text, and the fact that the manuscript mostly corresponds to the text supply additional constraints to enhance existing OCR software. Given accurate segmentation of lines, demarcated linguistic strings, and sample characters for training, partial OCR of the text may provide sufficient information to identify reliably and highlight the portion of an image that corresponds to a span of text. The known extent of text between confirmed termini can assist in segmenting subdomains and in locating passages in scanned images.

2 Work plan

The implementation of the project was planned over two years and one month from 1 June 2013 to 30 June 2015. A no-cost extension until 30 June 2016 was granted on 8 April 2015 to permit the principal software engineer on the project to attend to other duties. His attention was required to move our website to a commercial server and rebuild our website in a consistent format, and he desired leave to teach a couple of intensive one month courses in web security and recent developments in web design. Both distractions to his immediate responsibilities for the project funded by this grant indirectly benefited the project.

3 Transcription and OCR testing

The two project assistants at IIT Bombay, Anuja and Tanuja Ajotikar, prepared transcription of images of manuscripts of Sanskrit works written in Devanagari script using the transcription interface built by Ralph Bunker in the first year. Transcribed data included initially 7 pages of UPenn 2173, and then 1,063 pages from 142 mss. including additional pages from 2173. The two project assistants also produced 178 OCR files of 139 different manuscripts using Hellwig's Sanskrit Devanagari OCR software ind.senz to provide initial data for subsequent experiments.

4 Experiments

4.1 Generalized OCR software

Using images and their corresponding transcription, Donglai Wei trained the general OCR software OCRopus developed by Thomas Breuel and tested its ability to recognize the text of UPenn 2173 in the image. When tested on the very same line on which the manuscript was trained with 5,000 iterations, OCRopus recognized the text exactly. When trained on 20 different lines with 5,000 iterations, OCRopus produced results considered poor by any standards of readability. We later quantified those results using the Lewenshtein edit distance to length ratio. The ratio 0.457 in fact compares well with subsequent tests. Yet at first we dismissed these results as poor. Although we considered it possible that with a large amount of training data the results using generalized OCR software might improve, and considered another approach would be to focus on the identification of specific more easily recognizable characters, we decided first to attempt results with OCR software designed specifically for Sanskrit in Devanagari script.

4.2 Language and script specific OCR software

In October 2014, Scharf conducted an experiment on the efficacy of Oliver Hellwig's ind.senz Sanskrit OCR software. OCR was performed on four pages comprising 5,912 characters without, and then with, training, and the number of errors were counted. The number of errors in the untrained OCR was only minimally greater than the number of errors in the trained OCR (102 in untrained, 101 in trained) though the errors were less egregious with trained OCR. The time to correct errors in the untrained was 61:21 while in the trained it was 52:20. Training, OCR, and correction of an 846-page fairly clearly printed book was estimated to take 13.28 minutes per thousand characters and to cost about 0.56 euros per thousand, about the cost of double-keyed data entry.

A limitation was discovered in the ability to train the ind.senz software: one is limited to correcting errors where the software is unable to decide which character a unit is from among certain characters. However, one is not able to correct errors in OCR about which the software entertains no doubt in the first place. This problem is exacerbated in proportion to the deterioration of text quality. Since the text quality is poor in manuscripts because of natural variance in human writing even in neatly written manuscripts, the capacity of the ind.senz software to be trained for manuscript OCR becomes inadequate. As a consequence, we proceded to test with untrained OCR using ind.senz.

4.3 Base algorithm

In order to set the minimum standard for OCR alignment, we conducted an experiment to align images with corresponding digital text directly using no transcription or OCR results. We extracted all of the text from human-produced alignment data collected during a previous project to catalogue, digitize and align images of about 160 manuscripts at Brown University and the University of Pennsylvania with corresponding images. We took the beginning and endpoints of the corresponding digital text indicated in this data and distributed the text evenly over the number of images in the manuscript. We then measured the divergence of the actual number of characters per page from the estimated number of characters and produced graphs plotting the divergence. Figure 1 shows a graph of the results for UPenn 2252. The blue line shows the maximum number of characters exceeding the average by 910 characters on f. 45r, and the green line shows the maximum number of characters fewer than the average by 660 characters on f. 33r. Overall, the estimated location was never more than eight pages off and was an average of about three pages off. While this base algorithm vastly improves the ease with which a scholar would be able to locate a sought passage, this method merely established the minimum standard for our subsequent research.

4.4 Lewenshtein edit distance

For UPenn 492, a manuscript of the *Bhagavadgītā* consisting of 65 leaves or 130 pages, Bunker measured the Levenshtein edit distance between each line of the digital edition of the text and every segment of the same length in the OCR results of the manuscript and calculated the ratio of the edit distance to the length of the string. The smallest ratio represents the closest match. The images containing the closest matching lines were associated with that line of the edition. This procedure mapped 231 of the 519 lines of text (44.5%) to the correct image compared with the base algorithm which correctly mapped only 168 (32.4%).

4.5 Anchors

Bunker then introduced a procedure to locate mappings presumed to be correct and to redistribute the text between them. Where two consecutive lines were mapped to the same page they were assumed to be correct unless that mapping conflicted with another pair of consecutive mappings. Pairs of mappings were judged to conflict where they resulted in an inversion in the order of the text. The conflict was resolved by discarding the mapping pair whose position diverged most from the position predicted by the base algorithm. After resolving conflicts, the text was

redistributed evenly over pages between each set of anchors. This procedure carried out on UPenn 492 mapped 375 of the 519 lines of text (72.3%) to the correct image. The anchor procedure mapped 505/519 (97.3%) to within three pages compared with the base algorithm which mapped only 239/519 (46.0%) to within three pages. Figure 2 shows a table comparing results using the anchor algorithm with the edit distance algorithm and the base algorithm. Figure 3 shows one manuscript page, the digital text in the Sanskrit Library Phonetic ASCII encoding (SLP1), and the results of optical character recognition using Hellwig's ind.senz Sanskrit Devanagari OCR program converted to the same encoding.

4.6 Document layout and bibliographic references

The previous experiments utilized image-text alignment data produced by manually creating a correspondence between a region of an image and a passage of digital text using the Sanskrit Image Text Alignment program (SITA) developed by Scharf and Bunker in a previous project. We distributed text over the entire manuscript from the top of the page on which the first annotation occurred to the bottom of the page on which the last annotation occured without making any attempt to locate the start and end points within the page. We noticed that the greatest points of discrepancy occurred near the beginning and end and were likely due to the lack of precision in locating the position of the beginning and end text in the manuscript image. This affects not only the mapping of lines on the first and last pages, but also possibly the first and last lines on numerous pages near the beginning and end of the manuscript. A second short-coming of our first experiments was that they relied on alignment data previously created manually.

In a new set of experiments, we utilize data which, although created manually, is generally created while cataloguing a manuscript. Moreover, in The Sanskrit Library digital TEI-conformant XML catalogue, we also note the number of the line on which transcribed passages occur in the mansucript. This information permitted us to give greater precision to the beginning and end points of the text-image alignement and to collect information from a typical source — a manuscript catalogue — rather than from software specifically designed for manual image-text alignment.

The Sanskrit Library digital TEI-conformant XML catalogue entries include transcription of incipits and explicits and occasionally some additional text between, for example where missing leaves call for transcription of the surrounding text. These catalogue entries provide bibliographic references to editions, where available. Where digital editions of character data are available, the catalogue entries provide a `biblScope` element containing `from` and `to`, or `corresp` attributes that refer to lines in the digital text. The bibliographic elements are located

within note elements within the element containing the transcribed text correlated with the digital edition. Empty page break and line break elements in the text transcription area specify the page and line on which the text occurs in the manuscript. The correspondences provided by these location and bibliographic elements were used to refine our anchor algorithm to yield improved results in the alignment of the images with the digital edition.

For each text markup element in the catalogue entry that contained a biblScope element, we locate the preceding line break element (lb) and the preceding page break element (pb). The page break element facs attribute provides a reference to the xml:id of a graphic element referring to the image of the manuscript page in the facsimile section of the digital catalogue entry. The line break element provides an n attribute indicating the line in the manuscript image on which the following text begins. Page break and line break elements within the text markup element indicate additional page or line breaks within the transcribed text in that text markup element. The catalogue entry also provides a layout element writtenLines attribute that indicates the number of lines per page or the range of lines per page in the manuscript.

We use this information to calculate the approximate position of the beginning and end of the transcribed text in the manuscript image. The approximate beginning of the text on the page is the value of the n attribute of the preceding lb element divided by the average number of lines per page. The approximate end of the text on the page is the value of the n attribute of the last lb element within the text markup element divided by the average number of lines per page. The location of the passage within the manuscript is then given by the decimal value of the location within the page (of which there are generally two per folium, recto and verso) added to the page number minus one. For example, the writtenLines attribute in UPenn Ms. Indic 5 is 6. The first biblStruct element referenced by the incipit to the *bhhagavadgītā* occurs in an sp element whose preceding pb element refers to f. 2v and whose first preceding lb element refers to line 2. The last biblStruct element referenced by the explicit to the *bhhagavadgītā* occurs in an lg element the last pb element before the end of which refers to f. 129r and the last lb element before the end of which refers to line 5. The digital text of the *bhhagavadgītā* therefore begins on f. 2v line 2 and ends on f. 129r line 5. The number of pages covered by the text is 129 times 2 minus 1 (for the last verso) minus 3 (for ff. 1r–2r) which equals 254 pages diminished by 1/6 of the first and 1/6 of the last which equals 253.67. Numbering image pages from the one on which the text begins to the one on which the text ends, the text begins on image page (0+1/6=) 0.17 and ends on image page (254-1/6=) 253.83.

Having located the beginning and endpoints of text in the images, we then locate the beginning and end of the corresponding text in the digital edition. Bib-

liographic information provided by the `biblScope` attributes refers to a line of text in a digital edition. The number of lines in the digital edition between the start and end of the text in the manuscript is found by counting the lines beginning with the first line referenced by the first incipit `biblScope` element and ending with the last line referenced by the last explicit `biblScope` element. The `from` attribute of the first biblScope element of the incipit for the *bhhagavadgītā* in UPenn Ms. Indic 5 refers to BhG.slp#06023001 on line 7 of the digital edition, and the `to` attribute of the last `biblScope` element of the explicit refers to BhG.slp#06040078c on line 1461 of the digital edition for a total of (1461-7+1=) 1455 lines. The location of these lines in the manuscript is estimated by distributing those lines between the location of the initial bibliographic element and the location of the end of the final bibligraphic element in the digital images. For UPenn Ms. Indic 5, we distribute (1455/253.67=) 5.73 lines of text per manuscript page.

Although the number of lines to distribute per page is mathematically a fraction, we map whole lines. We distribute text rounding up until the fractional number of lines falls behind the distributed text by a whole number at which point we round down. We determine how many lines to distribute for each page between the first and last by multiplying the fraction estimating the number of lines per page (5.73) by the page number and subtracting the number of lines already mapped. UPenn Ms. Indic 5, we calculate the distribution of lines over the first several pages as follows:

5.73*1=	5.73	round=6
5.73*2=	11.46	round=12-6=6
5.73*3=	17.19	round=18-12-6
5.73*4=	22.92	round=23-18=5

Using the `pb` and `lb` elements differently places numerous lines at the beginning and end of the manuscript: line 5, 12, 17, 23, . . . , 157.

5 Results pending project completion

It is generally the case that rigorous systematic review of human-produced data turns up lacunae. The conduct of the experiment described in the previous section is no exception. We planned to use the text-image alignment data produced manually using our SITA program as the standard against which to compare the results of this experiment. However, when we compared the locations of text predicted by the `biblStruct` elements in our TEI digital catalogue entries with the locations of text in the human-produced SITA alignment data we found numerous instances where the SITA data was missing. In order to evaluate the results of the experi-

ment, we need to fix the data that serves as our gold standard. This is estimated to require a few months of additional work on the part of our assistants.

6 Data management

Bunker created an XML database in which to compile and store information regarding manuscript images, their transcription, OCR results, and correspondence with a digital edition. The database is accessed by software that automatically produces image-text alignment results and displays them in an HTML interface.

7 Outlook

We plan to conduct similar experiments with trained generalized software packages such as OCRopus and Tesseract. Bunker plans to create an interactive display of the manuscript page mappings that will permit human input into the alignment process to refine results. A scholar will be able to confirm mappings, create additional anchors, and redistribute the text between them in real time. We expect to be able to apply the text-image alignement software we have created to collections in which manuscripts are catalogued using the Sanskrit Library's TEI manuscript catalogue entries and are digitally imaged. The software will also be useful for any manuscripts so catalogued and imaged that have corresponding digital editions regardless of language.

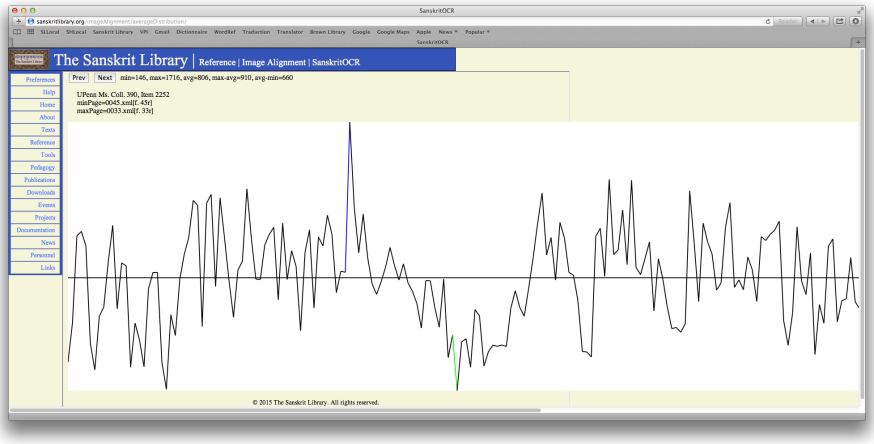
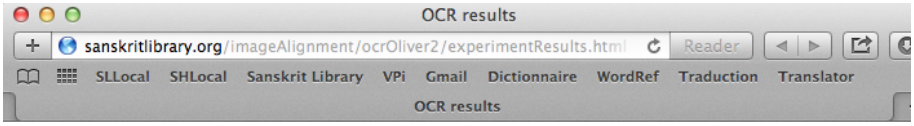


Figure 1: Divergence of the actual number of characters per page from the average in UPenn Ms. Coll. 390, item 2252.



Summarize

Extrapolations: 505 of 519 located within 3 images, 375 are exact
Estimates: 239 of 519 located within 3 images, 168 are exact
Average:0.60 **STD:**[0.50, 0.71] 404 ratios of 519 are within STD (163 are correct, 241 are wrong)

Line #	Correct image #(s)	Incorrect image #	Estimated image #	Extrapolated image #	Within std	matched dist/len=	matched ratio	missed dist/len=	missed ratio	
2	5	6	5	5		0/2	0	1/2	0.5	?
*3	5		5	5	*	38/60	0.63			?
4	5	16	5	5	*	8/16	0.5	11/16	0.69	?
5	5	20	5	6	*	15/28	0.54	16/28	0.57	?
*6	5		6	6	*	28/55	0.51			?
7	5,6	113	6	6	*	33/58	0.57	35/58	0.6	?
*8	6		6	7	*	26/57	0.46			?
9	6	7	6	7	*	41/68	0.6	43/68	0.63	?
*10	6,7		6	7	*	35/67	0.52			?
*11	7		7	7	*	39/70	0.56			?
*12	7		7	7	*	35/69	0.51			?
*13	7,8		7	7		28/70	0.4			?
*14	8		7	7		22/71	0.31			?

Figure 2: Summary of the comparison of the results of the edit-distance-anchor algorithm with the base algorithm and edit-distance algorithm

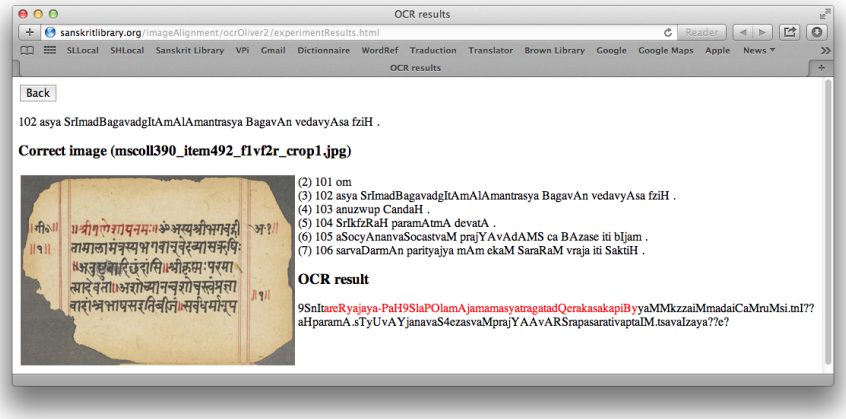


Figure 3: Image, digital edition text, and OCR results for UPenn Ms. Coll. 390, item 492, f. 1v