

Creating a Handwriting Recognition Corpus for Bushman Languages

Kyle Williams and Hussein Suleman

Department of Computer Science
University of Cape Town
Private Bag X3, Rondebosch, 7701
{kwilliams, hussein}@cs.uct.ac.za

Abstract. Handwriting recognition systems rely on the existence of a corpus for training recognition models and evaluating accuracy. Creating a handwriting recognition corpus for the Bushman languages of southern Africa is difficult due to the complexities of the script used to represent them and the fact that this script cannot be represented using Unicode. To solve this problem, a semi-automatic Web-based tool was developed to segment, capture and encode the Bushman text. A case study demonstrated how the tool could be used to create a Bushman handwriting corpus with few errors.

Key words: Corpus creation, transcription, digital libraries

1 Introduction

The Bleek and Lloyd Collection [1] contains notebooks that document Bushman language, culture and belief. These notebooks have been digitised and digital library systems have been built to preserve them and make them available online [1]. Transcriptions of the text that appears in the notebooks would allow for enhanced ways of interacting with the collection by allowing for the text to be indexed, searched and compared and could be used in speech-to-text applications and allow for the Bushman text to be reprinted in books. However, manual transcription is a time consuming and costly process and is generally not a viable option, thereby motivating the need for an automatic solution. In order to perform automatic transcription, a corpus is needed for training and evaluating automatic recognition systems. A problem arises in that the Bushman languages are complex due to the diacritics that they contain and cannot be represented using Unicode. To solve this problem, a custom Web-based tool was built that relied on automatic algorithms as well as user interaction to create a corpus.

The rest of this paper is structured as follows. Section 2 discusses some related work, followed by a description of the Bleek and Lloyd Collection in Section 3. The Web-based tool that was created to segment, capture and encode the Bushman text is described in Section 4 and a case study in Section 5 demonstrates the use of this tool to create a corpus. The quality of the data captured during the case study is analysed and evaluated in Section 6 and, finally, conclusions are drawn in Section 7.

2 Related Work

The existence of a high quality corpus is necessary for any system that performs automatic recognition of text, whether handwritten or machine printed. Corpora for English or other well-understood and well-studied languages are relatively easy to access and make use of. Examples of widely-used corpora are the Lancaster-Oslo/Bergen corpus [2] and the George Washington manuscripts [3]. However, for less well-studied languages and scripts, it is often necessary to create a new corpus that can be used for experimentation since it is unlikely that a suitable one already exists.

Creating corpora for modern languages is relatively easy compared to creating corpora for historical languages. Usually, creating corpora for modern languages involves getting a group of users to write words on forms that were specifically designed for creating the corpora. For instance, Agrawal et al [4] used HP TabletPCs to collect data and create a corpus for complex Indic scripts and Al-Ma'adeed et al [5] used paper forms on which specific Arabic words were written and then scanned to create an Arabic database.

The creation of corpora for historical texts is, however, more difficult than that of modern texts. For instance, it is not possible to make use of forms to easily and accurately capture specific data. Historical texts also introduce a number of difficult problems related to the segmentation of lines and words due to poor handwriting and the effects of age, such as ink-bleed and paper degradation. Fischer et al [6] note that, in many cases, transcriptions of historical texts can only be performed by language specialists, whereas for modern texts it can be performed by lay-persons. This is of course open to debate since it is possible to train lay-persons in transcribing the text or, alternatively, use a crowd-sourcing approach, which has, in many cases, been shown to be more accurate than specialists [7].

Fischer et al [6] describe a tool for the creation of a corpus for the IAM Historical Handwriting Database (IAM-HistDB). The specific manuscripts that they make use of all appear to have been scribe-written and are presented in a neat fashion. The tool described by Fischer et al [6] contains algorithms that automatically segment text on a page and that allows for users to manually correct any errors. The transcription of the text is not performed as transcriptions for the text already exist. Instead, transcription alignment is automatically performed and users are able to perform alignment correction.

Setlur et al [8] describe the complexities of the Devanagari script, which is the basis of many Indian languages and the lack of consistency among researchers in terms of the representation of the script, specifically in terms of what the granularity of a character should be. They describe a Java-based tool for creating a corpus for Devanagari script recognition, which automatically segments words and lines and also allows for user interaction. Once lines and words have been segmented, the Devanagari script is then captured using a simulated keyboard or by typing transliterated text on the keyboard. The text is stored in Unicode and the Devanagari output can be previewed by rendering the Unicode using a Devanagari font.

3 The Bleek and Lloyd Collection

The Bushman people are South Africa's oldest human inhabitants [9] and it is likely that they would have had a unique view of the world. However, over time much of their culture and language has been lost. Fortunately though, some of it was recorded and written down by linguists Wilhelm Bleek, Lucy Lloyd and others [1] in the 19th century and this historical collection of notebooks, dictionaries and pieces of art have collectively come to be known as the Bleek and Lloyd Collection [1]. Systems have and are being built for preserving the notebooks [1] and dictionaries [10] as well as for enhancing interaction with the collection [11]. An example of part of a notebook page is shown in Figure 1.

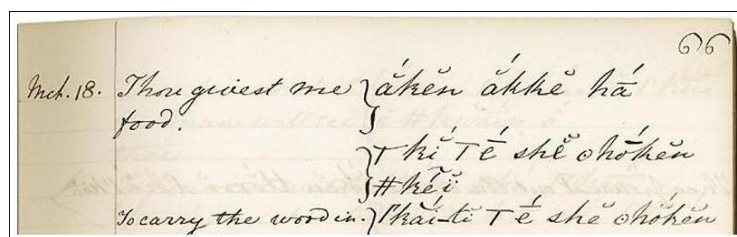


Fig. 1. An example of part of a page in the Bleek and Lloyd Collection

The script used to represent the Bushman text is complex due to the diacritics and the fact that the text cannot be represented using Unicode. Thus far, about 137 different combinations of single and stacked diacritics have been discovered that appear above, below and above and below characters. These diacritics need to be supported by a tool for capturing Bushman text. Table 1 shows some of the types of diacritics that appear in the text, starting with simple diacritics that appear above characters, then diacritics that appear above and below characters and then diacritics that are stacked and that span multiple characters.

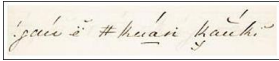

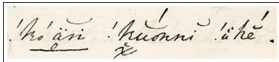
3.1 Bushman Text Representation

Since the Bushman text cannot be represented using Unicode, a custom solution for representing the characters was developed. The custom solution involves the use of \LaTeX and the TIPA package [12] for processing International Phonetic Alphabet symbols. The TIPA package does not support many of the diacritics that appear in the Bushman text by default. However, it does allow for the creation of custom macros that can be used to create nested and stacked diacritics.

Each Bushman diacritic is represented using a backslash (\backslash) followed by a command that specifies the diacritic and braces ($\{\}$) that contain the characters that the diacritics should be added to. Using the custom macros, the Bushman text can be encoded and represented. While the encoding is generally correct, the visual representation is just an approximation. Future work includes the

possibility of creating a custom font for Bushman script. Table 1 shows how Bushman text can be encoded using L^AT_EX and the custom TIPA macros and also shows the approximate visual representation.

Table 1. Encoding and visual representation of Bushman text using custom TIPA macros

Text Line	Encoding	Visual Representation
	<code>!ga\dialine{u} \twodots{e}</code> <code>\texthash{}ku\barblinet{a}n</code> <code>\ybelow{k}a\uline{u}k\{u}i</code>	<code>!gaú ã #kuani kaúki</code>
	<code>!k\uline{u} \twodots{i}</code> <code>y\dialine{a}ke\{u}n</code> <code>!ku\uline{i}-y\uline{a} .</code>	<code>!kú i yakeñ !kuí-ya' .</code>
	<code>!k\barbelow{\dialine{o}</code> <code>\circbtwodotst{a}\onedot{n}}</code> <code>\xcbelow{k}\uline{uo}nn\{u}i</code> <code>!\uu{u}h\uline{e} .</code>	<code>!kóän kúonni !úhé' .</code>

4 A Specialised Tool for Capturing Bushman Text

A custom tool called *xóä'xóä*, which means *to write* in the |xam Bushman language, was created to assist in creating the corpus for Bushman handwriting recognition. *xóä'xóä* is an AJAX Web-based tool that allows multiple users to assist in the creation of the corpus simultaneously. *xóä'xóä* contains a segmentation component that allows for a page of Bushman text to be segmented into individual lines and words using automatic segmentation algorithms and user interaction to correct errors and a text capture component that allows users to capture the Bushman text using a special input form.

xóä'xóä also acts as a job management and monitoring system. Each user has an account on the system, which assigns segmentation and transcription jobs and keeps track of the jobs that they have completed. *xóä'xóä* will be discussed in detail in this section, starting with a description of the preprocessing and segmentation steps and then a discussion of how the Bushman text is captured.

4.1 Preprocessing

Preprocessing involves preparing the notebook pages for segmentation. The first step is the separation of Bushman text from English text since they appear alongside in the notebooks (see Figure 1). The user is able to use a rectangle selection tool to draw a box around the Bushman text, which is then automatically cropped. In cases where the page contains no Bushman text, the user is able to click a button and the page will be ignored. The text is then thresholded

using a method based on a local adaptive approach [13] using a sliding window. Figure 2 shows an example of Bushman text that has been separated from the English text and thresholded.

4.2 Line Segmentation

To segment a page into the individual lines, an approach based on the horizontal projection profile is used [14] where each horizontal pixel-row is analysed to determine the number of foreground-background transitions and a horizontal projection profile is created. In the projection profile, minima represent the spaces between individual lines, which is smoothed using a Gaussian filter to remove false minima.

Before the page is segmented into lines, the candidate segmentation locations are presented to the user who is given the option to move them and add or delete segmentation locations. Furthermore, the user is able to change the size of the Gaussian kernel. Figure 2 shows an example of the interface when the line segmentation candidates are automatically identified by the line segmentation algorithm. The lines are segmented once the user has made any changes that they feel are necessary and the system keeps track of each page, the lines that it was segmented into and which user performed the segmentation.

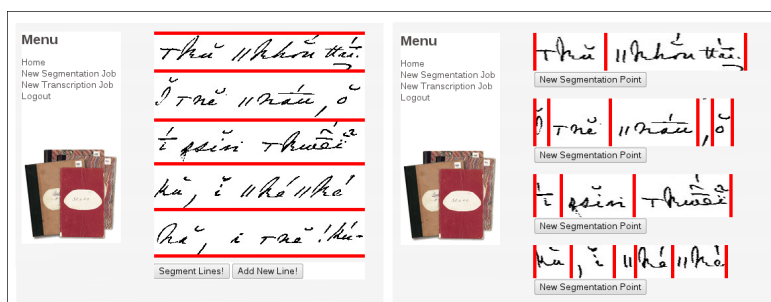


Fig. 2. Interface allowing user to view line and word segmentation candidates and change them

After the text lines have been segmented the slant is automatically corrected using a method based on the vertical projection profile [15]. The slant-corrected text lines are then presented to the user who has the option to crop them, re-apply the slant correction or even delete the line entirely.

4.3 Word Segmentation

Word segmentation is performed on each of the individual text lines segmented in the previous step using a technique based on the distance between connected components (CCs) [4]. In this method, the CCs in each line are first identified

and labelled [16] and then they are then sorted by their x-coordinate. The horizontal distances between every pair of adjacent CCs is calculated and when CCs overlap vertically the distance is set to 0. If the distance D between two adjacent CCs CC_i and CC_{i+1} is greater than the threshold T then CC_i and CC_{i+1} are considered as belonging to two separate words; otherwise, CC_i and CC_{i+1} are considered as belonging to the same word. The threshold T is calculated as:

$$T = \frac{\sum_{i=1}^{n-1} \text{Distance}(CC_i, CC_{i+1})}{n} / 2 \quad (1)$$

where n is the number of CCs in the image. As was the case with line segmentation, word segmentation candidates are presented to the user and the user has the option to move segmentation points as well as add and delete segmentation points. The words are segmented once the user is satisfied with the segmentation points and the system automatically keeps track of each word that a line was segmented into and the user who performed the segmentation. Figure 2 shows an example of the interface for word segmentation.

4.4 Text Capture

In order to capture the text, a custom input form was created and is shown in Figure 3. The text input interface is made up of several components. The first of these contains the image that is to be captured and is displayed at the top of the interface. Below this is the text input box where the user can input the text that they see, which can then be copied to the \LaTeX box below. From the \LaTeX box users are able to highlight text and add diacritics by clicking on the diacritics that they would like to add from the right side of the interface. The interface also allows the user to preview the \LaTeX representation and keeps track of each transcription and the user who captured it.

The interface caters for diacritics that appear above, below and above and below diacritics. The diacritics that it supports were pre-determined by scanning through a subset of the full Bleek and Lloyd Collection. However, it is possible that some diacritics might be encountered that the interface does not support. If this occurs, the user is able to mark a line as not being supported by the interface and at a later stage these lines can be reviewed and previously unseen diacritics can be added to the interface. Text lines can also be marked when users feel that they are not suitable for transcription.

5 Corpus Creation Case Study

A workshop was held in which twenty-nine data capturers used $x\acute{o}\grave{a}'x\acute{o}\grave{a}$ to assist in creating a corpus of Bushman handwriting. The workshop began with an introduction to the problem, a discussion of what the goals of the workshop were and what the data created would be used for and the data capturers were encouraged to talk to one another during the workshop. For the workshop a total of 900 pages in the collection, written by two authors, were randomly sampled and inserted into the job management system as segmentation jobs.

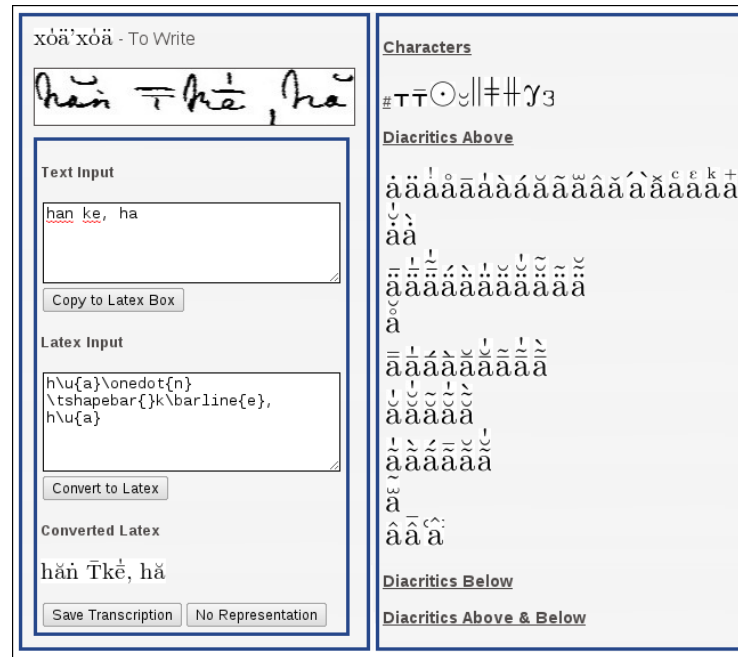


Fig. 3. Interface allowing user to capture the Bushman text

5.1 Segmentation

The segmentation part of the workshop started off with a short demonstration of how the segmentation was to be performed. Segmentation jobs were then randomly allocated to users by the job management system, which kept track of who the jobs were allocated to and what changes each individual user made. Text lines were the standard input method for capturing the Bushman text and therefore the segmented lines were added to a list of transcription jobs.

In total there were 900 segmentation jobs, of which 729 were completed and segmented into 7950 text lines. There are a number of reasons why the other 171 jobs might not have been completed, such as them not containing Bushman text. These incomplete jobs will be reviewed in a follow-up workshop.

5.2 Text Capture

Transcriptions were captured for each of the individual text lines. The data capturers were given a demonstration of how to perform transcription and the various special cases that arise in the Bushman text were highlighted. The data capturers were encouraged to collaboratively determine what the characters were and the importance of consistency was stressed.

Of the 7950 transcription jobs, 1547 were completed and 452 were marked as having no representation. Future work will investigate each of the text lines that

was marked as having no representation and appropriate action will be taken, such as adding the diacritics to the interface.

There are two reasons for 5951 transcription jobs not being completed. The first is that there was insufficient time to complete all of the jobs in one sitting and the other is that some users were more efficient at completing transcriptions.

6 Evaluation

The previous section described a case study where xóä'xóä was used to create a handwriting recognition corpus. In this section, the output from the case study is evaluated in terms of the quantity and quality of contributions of each data capturer and the relationship between them. This information is valuable in planning for a follow-on workshop to expand the corpus as well as to give an idea of the quality of the data produced during the workshop.

6.1 Contributions

The first issue considered is the contribution to the corpus made by each data capturer as it is important in planning for follow-on workshops. Table 2 shows the segmentation and transcription jobs completed by the data capturers and, as can be seen, there is wide variation between the different data capturers.

Table 2. Number of segmentation and transcription jobs completed by users

Segmentation Jobs		Transcription Jobs	
Jobs	Number of Users	Jobs	Number of Users
0-9	0	0-9	0
10-19	5	10-19	1
20-29	9	20-29	3
30-39	7	30-39	4
40-49	5	40-49	9
50-59	0	50-59	4
60-69	0	60-69	6
70-80	0	70-80	2

6.2 Data Quality

In the case of a corpus for handwriting recognition, quality is represented by the accuracy and correctness of the transcriptions of the data. To evaluate quality, three transcriptions by each data capturer were randomly sampled and reviewed by a post-graduate research assistant who noted the errors encountered in each text line. It was found that, on average, each text line transcription had 0.48 errors, where an error represents a character or diacritic that was transcribed

incorrectly. Given that most text lines contain more than 10 characters, it is felt that this number is acceptable for the purpose of handwriting recognition and shows that lay-persons are able to produce high quality data with relatively low levels of perceived errors.

An investigation was also conducted to see if there was a relationship between the number of transcriptions made by each user and the error rate. Figure 4 provides a comparison between the number of transcriptions and number of errors made by each data capturer and shows that there appears to be no relationship between the two.

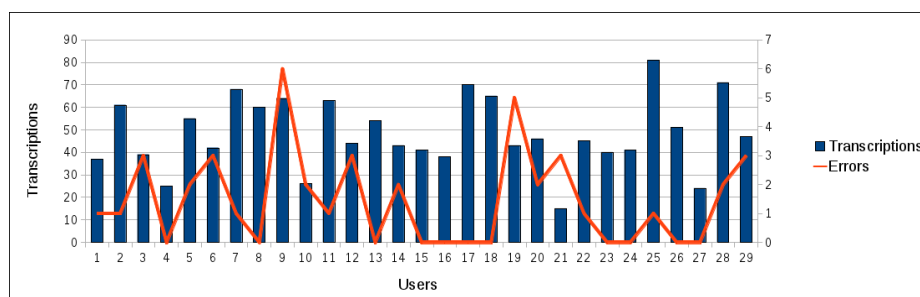


Fig. 4. Comparison of number of transcriptions and errors

The evaluation has shown how the different data capturers contributed to the creation of the corpus in terms of efficiency. It has also shown that accuracy and efficiency appear to be unrelated. Both of these findings are useful in planning for future corpus creation workshops.

7 Conclusions

The creation of corpora for historical texts is often difficult due to the effects of degradation and the complexities of historical scripts. This paper has discussed the creation of a corpus for the Bushman languages using a Web-based tool that was specifically built to cater for the complexities of the Bushman script. The tool was semi-automatic and involved automatic algorithms as well as operations that required user interaction. A case study demonstrated the use of the tool in a workshop to create a corpus and an evaluation of the data showed that data capturers were able to create a corpus with few perceived errors. The creation of transcriptions is a first step towards making the Bushman texts more accessible and it is hoped that it will allow for the development of new techniques for exploring and interacting with the collection.

Acknowledgements This research is supported by the University of Cape Town, the Telkom/NSN/Telescience/THRIP Centre of Excellence and the National Research Foundation.

References

1. Suleman, H.: Digital libraries without databases: The Bleek and Lloyd collection. In: Kovacs, L., Fuhr, N. Meghini, C. (eds.) *Research and Advanced Technology for Digital Libraries*. LNCS, vol 4675, pp. 392–403. Springer Berlin/Heidelberg (2007)
2. Marti, U., Bunke, H.: A full English sentence database for off-line handwriting recognition. In: *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pp 705-708, IEEE, Washington, DC (1999)
3. Rath, T. M., Manmatha, R.: Word spotting for historical documents. *Int. J. Doc. Anal. Recognit.* 9, 139–152 (2007)
4. Makridis, M., Nikolaou, N., Gatos, B.: An efficient word segmentation technique for historical and degraded machine-printed documents. In: *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp 178–182, IEEE, Washington, DC (2007)
5. Al-Ma'adeed, S., Elliman, D., Higgins, C. A.: A data base for arabic handwritten text recognition research. In: *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pp 485–489. IEEE, Washington, DC (2002)
6. Fischer, A., Indermühle, E., Bunke, H., Viehhauser, G., Stolz, M.: Ground truth creation for handwriting recognition in historical documents. In: *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp 3–10. ACM, New York (2010)
7. Surowiecki, J.: *The wisdom of crowds: why the many are smarter than the few*. Abacus (2005)
8. Setlur, S., Kompalli, S., Ramanaprasad, V., Govindaraju, V.: Creation of data resources and design of an evaluation test bed for Devanagari script recognition. In: *13th International Workshop on Research Issues in Data Engineering: Multilingual Information Management*, pp/ 55–61, IEEE, Washington, DC (2003)
9. Lee, R. A., Balick, M. J.: Indigenous use of *hoodia gordonii* and appetite suppression. *EXPLORE: The Journal of Science and Healing*. 3(4), 404–406 (2007)
10. Williams, K., Manilal, S., Molwantoa, L., Suleman, H.: A visual dictionary for an extinct language. In: Chowdhury, G., Koo, C., Hunter, J. (eds.) *The Role of Digital Libraries in a Time of Global Change*. LNCS, vol. 6102, pp. 1–4. Springer Berlin/Heidelberg (2010)
11. Williams, K., Suleman, H.: Translating handwritten bushman texts. In: *Proceedings of the 10th annual Joint Conference on Digital libraries*, pp. 109–118, ACM, New York (2010)
12. Rei, F.: Tipa: A system for processing phonetic symbols in \LaTeX . *TUGboat*. 17(2), 102–114 (1996)
13. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging*. 13(1), 146–168 (2007)
14. Marti, U., Bunke, H.: On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In: *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pp 260–265, IEEE, Washington, DC (2001)
15. Pastor, M., Toselli, A. H., Vidal, E.: Projection profile based algorithm for slant removal. In: Campilho, A., Kamel, M. (eds.) *Image Analysis and Recognition 2004*. LNCS, vol. 3212, pp. 183-190. Springer Berlin/Heidelberg (2004)
16. Shapiro, L., Stockman, G.: *Computer vision*. Prentice Hall (2001)