Title:

Collaborative Annotation and Computational Analysis of Hieratic

Author(s):

Julius Tabin, Mark-Jan Nederhof and Christian Casey

Collaborative Annotation and Computational Analysis of Hieratic

Julius Tabin[1][0000−0002−3591−6620], Mark-Jan Nederhof[2] (✉)[0000−0002−1845−6829], and Christian Casey[3][0000−0003−3423−5177]

[1] Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA
jtabin@g.harvard.edu
[2] School of Computer Science, University of St Andrews, KY16 9SX, UK
markjan.nederhof@googlemail.com
[3] Freie Universität Berlin, Department of History and Cultural Studies, 14195 Berlin, Germany
christiancasey86@gmail.com

Abstract. We introduce a new open-source web application for the collaborative annotation and study of hieratic texts, in the form of facsimiles and/or photographs. Functionality includes the automatic classification of occurrences of hieratic signs in Unicode, aided by image processing and optical character recognition techniques. Relying on various forms of dimensionality reduction, the tool also allows visualization of differences in sign shapes between texts, periods, genres, and geographical areas. This is motivated by recent work that demonstrated the value of dimensionality reduction for such purposes as identifying scribal hands. The interactive user interface can also be used for teaching students to read hieratic.

Keywords: Hieratic · OCR · Dimensionality reduction · Visualization

1 Introduction

There are currently a number of databases that gather photographs or facsimiles of texts, with the aim of facilitating palaeographic research, that is, investigations into the shapes of glyphs and the possible conclusions that can be drawn from these shapes about dating and geographical areas [24], scribal hands [16], writing materials [17], and evolution of scripts [19]. In most cases, there is a substantial gap between the breadth of the collected data and the computational analysis of the shapes. Bridging this gap typically involves manual selection of a few tokens that are deemed to be of interest. This limits the kinds of computational analysis that can be done, as much of the variation is lost in the selection.

One example of a project that does exhaustive labeling of glyphs is *the Demotic Palaeographical Database Project.*[4] However, it merely excises areas around

[4] http://www.demotischdemotisch.de/

glyphs, without representing the shapes of the glyphs explicitly. Similarly, *DigiPal*[5] and *HebrewPal*[6] merely encode rectangles around selected occurrences of letters in photographs of texts, and moreover do not exhaustively annotate any texts. The *ICDAR2023 Competition on Detection and Recognition of Greek Letters on Papyri*[7] comprised an exhaustive annotation of many dozens of texts in the MS COCO format [8], and encoded rectangles around occurrences of letters.

In contrast to all of the above, the *Paläographie des Hieratischen and der Kursivhieroglyphen*[8] collects the shapes of (selected) glyphs, extracted from facsimiles. However, at least at present, the project does not exhaustively annotate any texts.

In the case of alphabetical scripts, there is little variation in letter shapes in any given text, and the added value of exhaustive annotation, as opposed to selection of a few instances, has limited value to palaeography. However, Ancient Egyptian has many more signs, which each occur with smaller relative frequency and with greater variation. This suggests that exhaustive annotation holds greater potential for Egyptological studies than in the case of alphabetical scripts. This holds particularly well for the hieratic script, which is notable for its ligatures [4, 20], where multiple glyphs are joined together, further extending the number of unique signs to be categorized. Hieratic lends itself relatively well to a digital representation in grayscale and to image processing techniques. However, with off-the-shelf tools, exhaustive annotation of the shapes of hieratic glyphs requires a prohibitive amount of manual labor.

The first contribution of this paper is a demonstration that exhaustive annotation of hieratic facsimiles is feasible with our customized tool, called *Isut*, which is implemented as a web application.[9] After explaining the structure of the data in Section 2, Section 3 describes the user interface for annotating a text, and Section 4 presents several ways to check annotations for accuracy. Section 5 summarizes the applied technology.

The second contribution, addressed in Section 6, is to demonstrate that exhaustively annotated texts allow analyses that would be impossible with sparsely annotated texts where only a very limited number of tokens per type would be collected. This builds on [22], which explored the use of UMAP dimensionality reduction for making inferences about scribal hands, for automatic discovery of distinct hieratic types of the same underlying hieroglyphic sign, and for identifying distinct signs whose graphical realizations can be easily confused. Section 7 then provides an example of specific hypotheses that can be tested against the available data.

Lastly, Section 8 presents different scenarios of use for collaborative annotation and addresses potential applications.

[5] http://www.digipal.eu/
[6] https://www.hebrewpalaeography.com/
[7] https://lme.tf.fau.de/competitions/2023-competition-on-detection-and-recognition-of-greek-letters-on-papyri/
[8] https://aku-pal.uni-mainz.de/
[9] From , *sš (n) jswt*, 'writings of the ancient ancestors'.

2 Data

The data of Isut consists of two parts: a traditional database and a directory structure of image files. There is one database record for each text, consisting of:

- (unique index,)
- name,
- creator (of the facsimiles),
- provenance (if known),
- period (typically a dynasty or range of dynasties),
- genre,
- notes (such as copyright information),
- edit history (list of usernames and dates),
- zero or more pages.

For each page, the record contains:

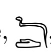
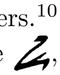
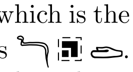
- (unique index,)
- type of the main image of the page (generally 'facsimile'),
- zero or more alternative images (e.g. photographs),
- zero or more lines (or columns).

For each line (or column), the record contains:

- (unique index,)
- line number,
- polygon around the line (in terms of coordinates in the main image of the page),
- direction of writing (either 'horizontal' for a line of text or 'vertical' for a column),
- zero or more glyphs.

For each glyph, the record contains:

- (unique index,)
- Unicode string identifying a hieroglyphic sign or a combination of signs,
- coordinate of the top-left corner of the bounding box, in terms of pixel positions in the main image of the page, as well as the width and height measured in the number of pixels.

A Unicode string may include control characters.[10] For example, , which is the normalized representation of hieratic ligature , can be encoded as . The limitations of normalized hieroglyphs for encoding hieratic signs have been

[10] Unicode 15, introduced in September 2022, brings the total number of control characters for Ancient Egyptian to 17, in combination with a number of modifying and other special characters [5].

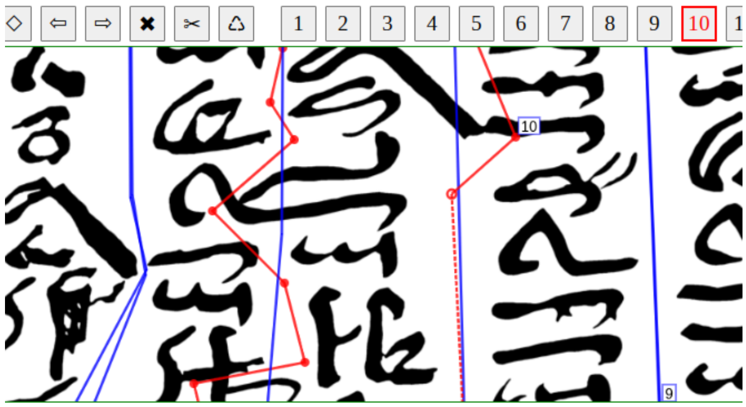


Fig. 1. Polygons delimiting columns of text, which may overlap due to 'tails' of signs protruding into neighboring columns.

acknowledged in the literature, and more fine-grained encodings have been proposed that were designed specifically for hieratic [12, 6]. Such encodings can differentiate between hieratic forms that correspond to the same hieroglyphic sign. Our motivation for a more traditional and coarse-grained encoding is that it offers the best prospects for stability in the medium term, and allows us to explore how different hieratic forms can be discovered with the help of computational techniques.

The image files are arranged into a directory structure, with file and directory names identified by the unique indices referred to above. For each line, there is a directory gathering all grayscale image files of glyphs in that line. These images have a transparent background, so they can be overlaid on top of the facsimile or photograph. For each page, there is a directory containing the facsimile and possibly a photograph or other additional images, and subdirectories for the individual lines. For each text, there is a directory that has one subdirectory for each page.

3 Annotation

Exhaustive annotation of texts with occurrences of hieratic signs is facilitated by a number of customized graphical user interfaces. The first step in the annotation of a page is to delineate lines (or columns) of text. This is primarily done by a series of mouse clicks on the image of the facsimile, to form a polygon. The polygon is intended to encompass all glyphs belonging to a line, but may include parts of glyphs belonging to neighboring lines. It is quite frequent in hieratic that a 'tail' of a sign in one line protrudes into a neighboring line, as exemplified in Figure 1.

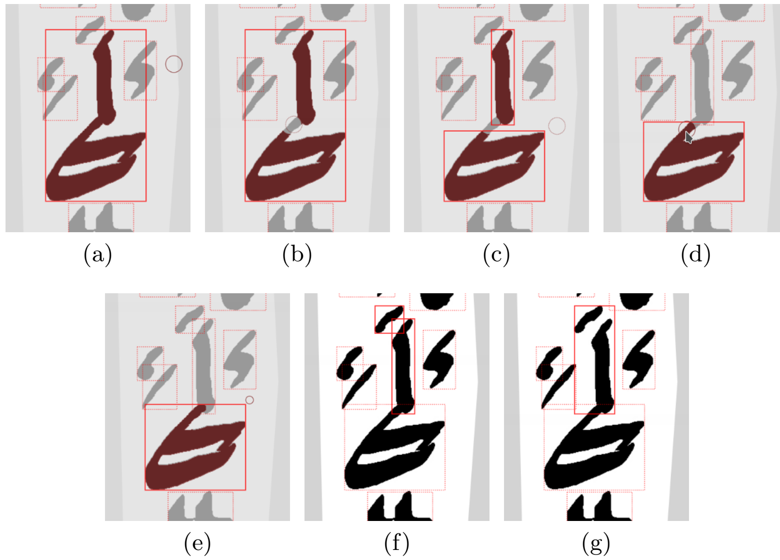


Fig. 2. (a) A blob (within the solid red rectangle) contains parts of two signs. (b) Pixels connecting the two signs are erased, leaving the upper sign intact. (c) The selected blob is split into two new blobs. (d-e) The erased pixels from the lower glyph are restored. (f) Two blobs are selected (in solid red boxes). (g) These are then merged into one glyph.

Each polygon around a line represents an excised part of a page, which is presented in a second type of user interface. In the most typical case, the next step in the annotation is to invoke the program's ability to automatically identify all "blobs" in the line. A *blob*, also called a *connected component*, is a maximal collection of black (or, in general, dark) pixels that are connected to one another. Blobs that are smaller than a user-defined threshold are ignored.

Initially, the candidate glyphs in a line are the automatically obtained blobs. However, hieratic signs may consist of several blobs and one blob may consist of several hieratic signs. In the latter case, the blob may be a generally recognized ligature or it may be the result of an accidental touching or crossing of glyphs, even between neighboring lines, as exemplified earlier. To deal with these cases, a number of glyphs may be selected and merged into one glyph or a glyph may be split into several by first erasing pixels from a blob by an 'eraser' tool, effectively 'cutting up' a blob into multiple blobs. The separated glyphs are then 'repaired' by filling the pixels back in that were previously erased, by a 'paint' tool that only paints pixels that are dark in the facsimile. Figure 2 illustrates this process.

Another tool can be used to fill in patches of light pixels surrounded by dark pixels. This can help annotate facsimiles where glyphs in red ink are suggested

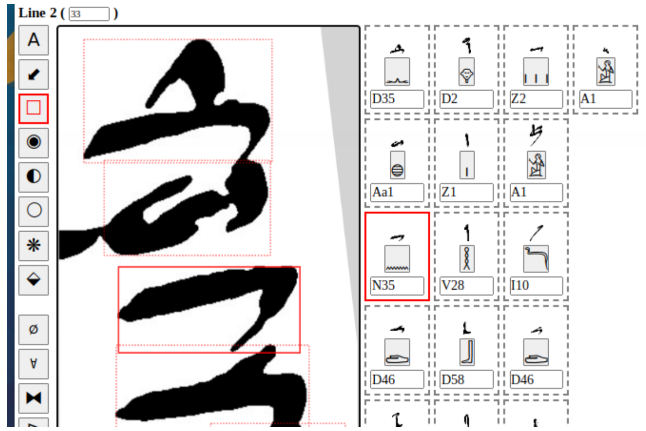


Fig. 3. Glyphs (left) with their labels (right)

by drawing only the outlines. The tool then effectively fills in the space between the outlines.

In the final stage of annotation, the candidate glyphs are classified, that is, they are labeled with a Unicode string of normalized hieroglyphs, possibly including control characters. First, each glyph is given a label automatically, which can then be manually corrected by entering the correct name or through the HieroJax graphical Unicode editor.[11] The view at the end of this process is exemplified by Figure 3.

Automatic classification of glyphs is implemented as follows. Glyphs are scaled to 16 by 16 bilevel grids and the set of such grids is subjected to principal component analysis (PCA). For each glyph, a vector consisting of the first 40 principal components is stored in a table, together with its ground-truth label. For a new glyph to be classified, a new vector is calculated in the same way, which is then compared to the vectors in the table, in terms of Euclidean distance. The label belonging to the nearest vector is returned. The reason we have opted for PCA is that it is very fast and runs on any platform without dedicated hardware, such as GPUs. Moreover, PCA does not require large amounts of training data, which can be a bottleneck for hieratic, as there can be relatively many types and few tokens per type. A single sign occurrence seen before may suffice to recognize a new occurrence of the same sign.

To be able to report an objective measure of accuracy, we have tested our classifier for individual signs. We excluded ligatures because opinions vary when a pair (or triple, . . .) of overlapping or touching hieratic signs should be seen as a ligature. This left 7893 tokens and 325 types. These were randomly shuffled. We then set aside a quarter (1973 tokens) for testing and applied PCA on the remainder (5920 tokens) to build the table. For each test token we then

[11] https://github.com/nederhof/hierojax.

determined the type of the nearest token in the table, in terms of the Euclidean distance, which resulted in an accuracy of 73%. (In over 90% of the cases, the 5 types with the nearest tokens include the correct type; this is over 92% for 10 types.)

Our 73% accuracy may seem lower than the 85% accuracy reported by [3] for a 8k dataset. However, that dataset had only 39 types, an order of magnitude below our 325 types, and it seems unlikely that their experimental conditions were representative. We found that the 39 most common types only account for 77.5% of the tokens.

There is a stark contrast with alphabetical scripts for which many tokens for relatively few types are available for training. This allows effective parameter estimation of neural networks, which can for some applications reach almost perfect accuracy [1]. Somewhat more similar to hieratic may be Chinese handwriting, for which accuracies above 90% have been reported [25]. However, this required two orders of magnitude more training data than was available for our experiments. A fair comparison is therefore hard to make.

The annotation process has been extensively tested by the first two authors. Facsimiles of two texts (the Shipwrecked Sailor and the Eloquent Peasant B1) were previously prepared by the first author as part of the work described in [22], and were annotated by the second author. The remaining facsimiles were taken from [14] and were annotated by the first author. Labeling of glyphs was done on the basis of published hieroglyphic transcriptions. We found that the time needed for annotation can vary depending on the degree of overlap of glyphs, as the procedure of "cutting up and repairing", as discussed earlier, is the most time-consuming part of annotation. If almost no glyphs overlap, annotation of a single line typically takes on the order of two minutes. If there is more overlap, then it can take up to 10-15 minutes per line. We hope to do systematic user studies at a later stage.

4 Validation

In the annotation of large numbers of facsimiles, it is inevitable that mistakes are made. The application offers a number of views of the data that help identify and correct such mistakes. First, all occurrences of glyphs with the same label (normalized transcription as hieroglyphs) can be listed and viewed next to each other. The glyphs can also be overlaid on a slightly bigger square excised from the facsimile, to see the glyphs in context; see Figure 4.

Second, dimensionality reduction (cf. Section 6) can reveal outliers in shapes, which may indicate errors.

Lastly, a view of an annotated facsimile is provided that shows glyphs in different colors, such that no two glyphs can have the same color if their bounding boxes overlap or are close to one another, as illustrated by Figure 5. A distinct color results where two glyphs overlap, unless the mouse hovers over a glyph, in which case that glyph's color supersedes the colors of the other glyph(s), as shown by Figure 6. In this way, one can see at a glance whether any ink has

Fig. 4. Listing of glyphs for 𓁷 (D2), with surrounding context.

remained unaccounted for, or if ink has been attributed to the wrong glyph. In addition, the labels of the glyphs are shown at a fixed distance from the glyphs themselves.

5 Implementation

Isut is a Node.js web application, and has been tested on Linux and macOS. The database technology is MongoDB. Source code and data are available on GitHub.[12]

Dimensionality reduction and OCR were implemented in Python, running on the server. This is motivated by the existence of standard Python packages that offer the required functionality and are highly optimized for speed. The latency caused by communication between client and server for the purposes of dimensionality reduction and OCR is not likely to be a hindrance.

In contrast, functionality to annotate pages and lines is done completely client-side. For example, the user interface to add points to a polygon that delineates a line is implemented in HTML canvas and JavaScript responding to mouse clicks and mouse movements. Also, recognition and manipulation of blobs are implemented in client-side JavaScript. In this way, web pages remain responsive even in the case of slow internet connections. Modified annotations are asynchronously sent to the server every few seconds, to minimize the risk that work is lost if there is any disruption to connectivity.

6 Analysis

Dimensionality reduction turns high dimensional data into a form that is more amenable to analysis and visual representation, with the potential to gain new insights. Isut applies dimensionality reduction on representations of glyphs as

[12] https://github.com/nederhof/isut.

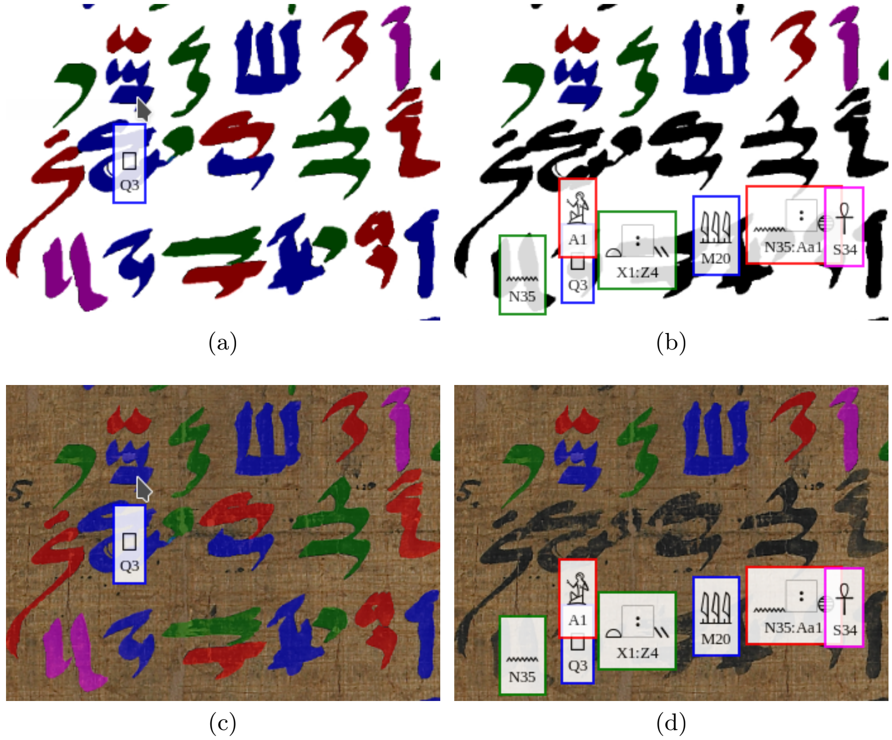


Fig. 5. (a) In one view, different glyphs are distinguished by color, and by hovering the mouse over a glyph, its label is shown. (b) In another view, labels of all glyphs of a selected line are shown. (c-d) The glyphs can also be superimposed on a photograph.

vectors of length $16 * 16$. Such a vector is obtained by first rescaling a glyph to a grid of 16 by 16 pixels, and then converting the image to bilevel.

The following methods are currently available:

- principal component analysis (PCA) [7]
- t-distributed stochastic neighbor embedding (t-SNE) [9]
- multidimensional scaling (MDS) [11]
- Isomap [23]
- spectral embedding [2]
- locally linear embedding [21]
- uniform manifold approximation and projection (UMAP) [10]

With each of these methods, the user can opt for reduction to 1, 2, or 3 dimensions. The default is PCA to 2 dimensions, which often gives the best results, exemplified by Figure 7. This simple example deals with one sign represented in two texts, and, for each token, a circle appears with a color that depends on the text. In general, one can include any number of signs and any number of texts,

(a) (b) (c)

Fig. 6. (a) Overlap of glyphs is shown in a color distinct from the colors of the individual glyphs. (b-c) By hovering the mouse over a glyph, it is raised over neighboring glyphs.

and can group texts together into periods, genres, or geographical areas, to, for example, obtain one distinct color for each period. By hovering over a circle, the shape of the glyph is shown; by simultaneously pressing the space bar, the shape is added to one of the corners of the graph, as illustrated here. Clicking on a circle opens a page showing the facsimile, with the selected occurrence of the glyph marked by its bounding box.

When viewing the output from dimensionality reduction to 3 dimensions, the user can use the mouse to rotate the model and inspect it from different angles. Points closer to the 'camera' appear larger, as shown in Figure 8. Dimensionality reduction to 1 dimension is illustrated by Figure 9.

7 Case Study

Isut's aforementioned dimensionality reduction techniques have utility beyond the overall visualization of the data and its many didactic functions. Substantial cross-textual insights have previously been uncovered using dimensionality reduction [22], and Isut builds on this by making its repertoire of techniques quickly accessible and efficient. In this way, a researcher can search the data to make meaningful inferences about texts or illuminate trends in sign development and evolution.

For example, when dimensionality reduction is used to plot the data for sign 𓂋 (D21), the majority of the examples of the sign from the later time periods (Dynasties 15 and 18) cluster together at one end of the graph, indicating similarity (Figure 10). The 12th Dynasty glyphs that cluster with them are those from Texte aus Hatnub, which can be disregarded here, given that they were carved, rather than written. There is partial overlap, with some tokens from Dynasties 15 and 18 clustering with the tokens from Dynasties 12 and 13, but, nevertheless, the trend is clear. Once a hypothesis has been generated from observation of the data (e.g., the form of sign D21 changed over time, with a

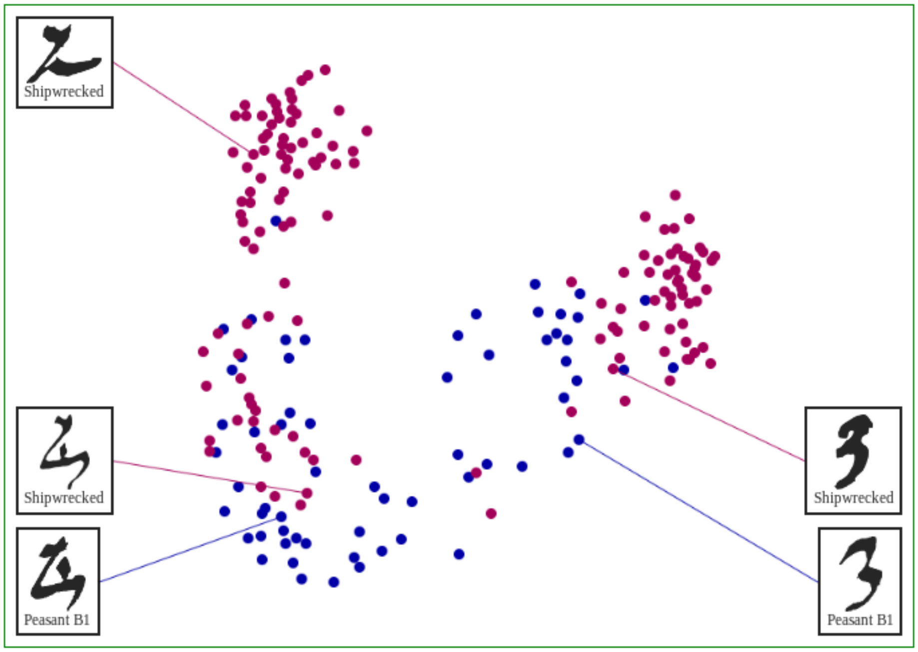


Fig. 7. PCA identifies three types of 𓅃 (G17) in the combination of the Shipwrecked Sailor and the Eloquent Peasant (manuscript B1). One of the types is found in Shipwrecked and Peasant B1 (bottom left). Another is found in Shipwrecked and Peasant B1 (bottom right). A third type is almost solely found in Shipwrecked (top left).

distinct form dominating the corpus in the later dynasties), Isut's mouse-over function, allowing individual tokens to be seen relative to their position on the graph, provides the potential to visualize these trends with unprecedented ease.

The selected tokens in Figure 10 are representative of the tokens around them. From such inspection, one can see that the later form of D21, shown here using individual signs from Papyrus Rhind and Westcar, is much shorter than the Dynasty 12/13 version, in both width and tail. In the earlier dynasties, there was already a smaller D21 by width, presented here using a glyph from Peasant B1, but, as can be seen, this is morphologically distinct from the later version of the sign, particularly with respect to tail length. This is a promising avenue for further research, since knowledge of the evolution of such a common and integral glyph can further our understanding of the physical development of the language as a whole. While this is not a new area of research (indeed, Möller [13] was categorizing the development of hieratic characters over 100 years ago), never before have researchers been able to view and compare hieratic data on this scale.

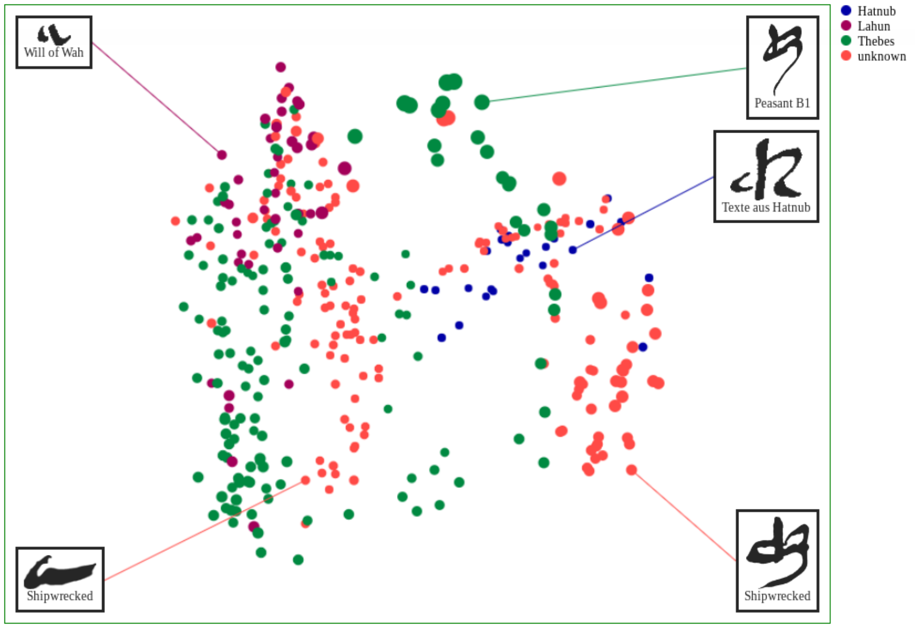


Fig. 8. Dimensionality reduction with Isomap for 𓀀 (A1) in 3 dimensions, per provenance.

Using this same functionality, Isut also allows hypotheses to be tested. For instance, if sign D21 truly changed form over time, one might predict that the often nearly indistinguishable sign ⏜ (X1) would have undergone a similar transformation; by form alone, the two hieratic signs are often indistinguishable to humans, as well as to OCR methods. With a single button press, a user can generate a corresponding dimensionality reduction graph for X1 in seconds (Figure 11). As expected, the X1 signs from Dynasties 15 and 18 largely group together, apart from the earlier dynasty material. As with D21, hovering over individual points reveals the trend: once again, there is a truncation of the tail of the sign. Thus, D21 and X1 changed in largely the same way, adding support for the idea that there was a distinct historical shift in the writing of these signs. By checking Isut's graphs for other signs, one can confirm that the later dynasty material does not always cluster together apart from the rest of the data (data not shown). This demonstrates that some signs, such as D21 and X1, were subject to greater change than others in the time between Dynasties 12 and 18. Investigations into these varying evolutionary rates of sign morphology is an important area for future work, which can be facilitated using this application.

Of course, these results will only become clearer as more data is added to Isut. Currently, the uploaded 18th Dynasty texts are restricted to Papyrus Ebers and Papyrus Westcar and the only 15th Dynasty text is the Rhind Papyrus. While the

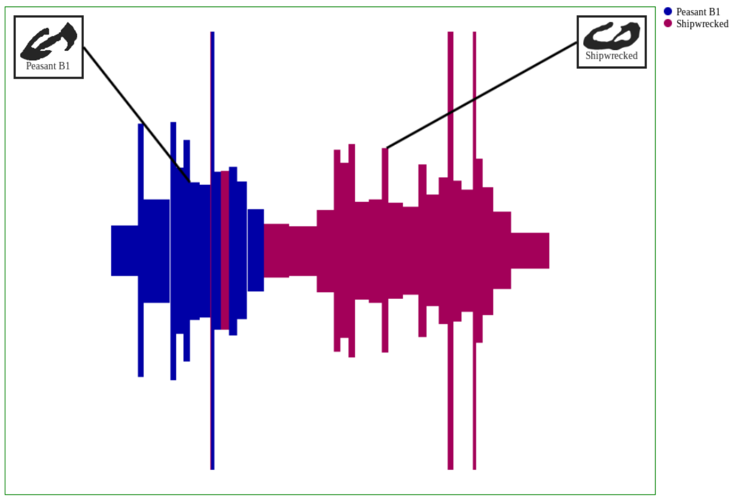


Fig. 9. PCA to one dimension for 𓐍 (Aa1), rendered in a two-dimensional diagram. Tokens are positioned from left to right according to the one dimension, but each token is represented by a rectangle with a fixed area, such that the rectangles of neighboring tokens touch but do not overlap. Thereby, clusters of tall rectangles suggest tokens that are close together. This graphic shows that the shapes of Aa1 in the Eloquent Peasant (B1) can be almost perfectly separated from those in the Shipwrecked Sailor.

above D21/X1 trend is visually obvious given the graphical output, the nuances of the morphological shift cannot be fully ascertained without a greater amount of data. For instance, it is conceivable that a new text from a later dynasty could be added to Isut's corpus containing D21/X1 signs with long tails, changing the overall interpretation from "D21/X1 were shortened over time" to "D21/X1 became more variable over time". The value of this tool, beyond its accessibility and power for large-scale research, lies in its adaptability, continually growing with more information and allowing an ever-expanding number of increasingly significant comparisons to be made.

8 Deployment

The current implementation can be deployed centrally or locally. Central deployment requires a publicly accessible web server. One or more accounts can be created for 'editors'. An editor can create accounts of 'contributors' or other editors and assign preliminary passwords. Both editors and contributors can add and modify texts and annotations. Local deployment can dispense with user administration and credentials.

Texts and their annotations can also be easily exchanged. From one instance of the application, a text consisting of its database record and image files can

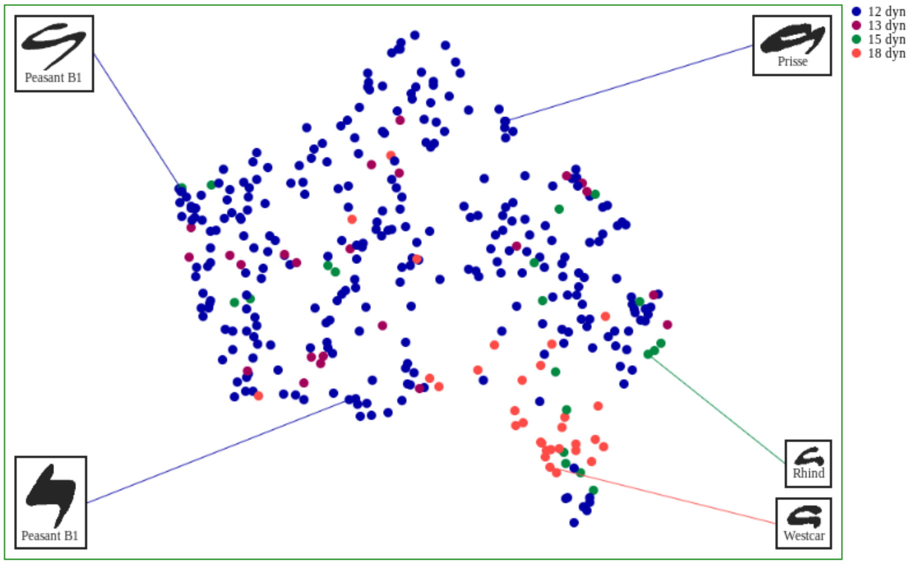


Fig. 10. UMAP graph for 𓂋 (D21), by period.

be downloaded into a single zip file. This zip file can be uploaded to another instance of the application. This allows for distributed creation of annotated texts, which, if desired, can be merged centrally, possibly after validation by moderators.

At present, material for teaching hieratic mainly consists of printed books. This includes [13, 14], which regrettably do not cover more recently discovered papyri. Other material is more up to date [15], but is in most cases still only available in printed form. There are good grounds to develop teaching material for hieratic in digital form [18], which can be more easily expanded and corrected on the basis of fresh evidence. Isut creates new opportunities for teaching hieratic, as new manuscripts can be added with relative ease, and the organization of the data allows development of tools for training and assessing hieratic reading skills.

9 Conclusions

We have introduced Isut, a tool to exhaustively annotate hieratic texts. The resulting data creates new potential for analysis and teaching of hieratic. As the data set expands, the power of analyses will grow and new insights will be able to be gained. Thus, the focus in the immediate future will be on expanding the collection of data, which currently comprises 691 distinct signs, and 8880 sign occurrences altogether, from thirteen texts from the 12th to the 18th Dynasty, covering three geographical areas and seven genres.

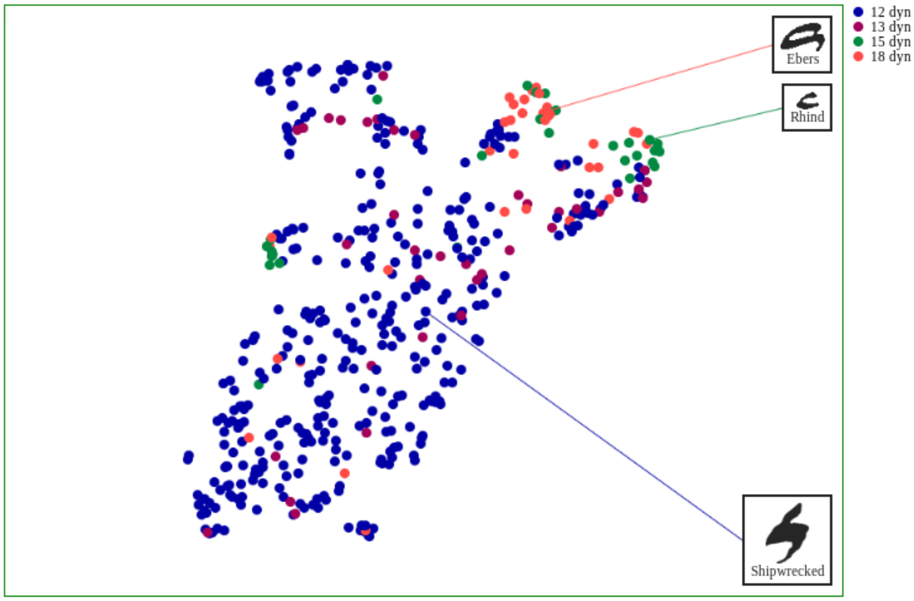


Fig. 11. UMAP graph for 𓏏 (X1), by period.

References

1. Ali, S., Abdulrazzaq, M.: A comprehensive overview of handwritten recognition techniques: A survey. Journal of Computer Science **19**, 569–587 (2023). https://doi.org/10.3844/jcssp.2023.569.587
2. Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.F., Vincent, P., Ouimet, M.: Learning eigenfunctions links spectral embedding and kernel PCA. Neural Computing **16**, 2197—2219 (2004). https://doi.org/10.1162/0899766041732396
3. Bermeitinger, B., Gülden, S., Konrad, T.: How to compute a shape: Optical character recognition of hieratic. In: Gracia Zamacona, C., Ortiz-García, J. (eds.) Handbook of Digital Egyptology: Texts, chap. 7, pp. 121–138. Editorial Universidad de Alcalá (2021). https://doi.org/10.25358/openscience-6757
4. Gasse, A.: Les ligatures dans les textes hiératiques du Nouvel Empire (à partir des ostraca) – entre pragmatisme et maniérisme. In: Gülden, S., van der Moezel, K., Verhoeven, U. (eds.) Ägyptologische "Binsen"-Weisheiten III, pp. 111–133. Franz Steiner Verlag (2018). https://doi.org/10.25358/openscience-403
5. Glass, A., Grotenhuis, J., Nederhof, M.J., Polis, S., Rosmorduc, S., Werning, D.: Additional control characters for Ancient Egyptian hieroglyphic texts. https://www.unicode.org/L2/L2021/21248-egyptian-controls.pdf (2021)
6. Gülden, S.: Paläographien und hieratogramme — digitale herausforderungen. In: Gülden, S., van der Moezel, K., Verhoeven, U. (eds.) Ägyptologische "Binsen"-Weisheiten III, pp. 83–109. Franz Steiner Verlag (2018). https://doi.org/doi.org/10.25358/openscience-390
7. Jolliffe, I.: Principal Component Analysis. Springer, New York, 2nd edn. (2002). https://doi.org/10.1007/b98835

8. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol. 8693, pp. 740–755. Zurich, Switzerland (2014). https://doi.org/10.1007/978-3-319-10602-1_48
9. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579—2605 (2008)
10. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. https://arxiv.org/abs/1802.03426 (2020)
11. Mead, A.: Review of the development of multidimensional scaling methods. Journal of the Royal Statistical Society Series **D 41**, 27–39 (1992). https://doi.org/10.2307/2348634
12. van der Moezel, K.: On signs, lists and standardisation. In: Gülden, S., van der Moezel, K., Verhoeven, U. (eds.) Ägyptologische "Binsen"-Weisheiten III, pp. 51–81. Franz Steiner Verlag (2018). https://doi.org/10.25358/openscience-391
13. Möller, G.: Hieratische Paläographie. Otto Zeller, Osnabrück (1909-1936), three volumes
14. Möller, G.: Hieratische Lesestücke für den akademischen Gebrauch. J.C. Hinrichs (1927-1935), three volumes
15. Möschen, S.: Lernen leicht gemacht? Arbeiten an einer Chrestomathie des Hieratischen. In: Gülden, S., van der Moezel, K., Verhoeven, U. (eds.) Ägyptologische "Binsen"-Weisheiten III, pp. 319–327. Franz Steiner Verlag (2018). https://doi.org/10.25358/openscience-405
16. Papaodysseus, C., Rousopoulos, P., Arabadjis, D., Panopoulou, F., Panagopoulos, M.: Handwriting automatic classification: Application to ancient Greek inscriptions. In: 2010 International Conference on Autonomous and Intelligent Systems. pp. 1–6. Povoa de Varzim, Portugal (2010). https://doi.org/10.1109/AIS.2010.5547045
17. Piquette, K., Whitehouse, R. (eds.): Material Practice: Substance, Surface and Medium. Ubiquity Press, London (2013). https://doi.org/10.5334/bai
18. Polis, S.: Methods, tools, and perspectives of hieratic palaeography. In: Davies, V., Laboury, D. (eds.) Oxford Handbook of Egyptian Epigraphy and Paleography, chap. IV.3, pp. 550–565. Oxford University Press (2020). https://doi.org/10.1093/oxfordhb/9780190604653.013.39, online edition
19. Rajan, V.: How handwriting evolves: An initial quantitative analysis of the development of Indic scripts. In: 17th Biennial Conference of the International Graphonomics Society. Pointe-à-Pitre, Guadeloupe (2015)
20. Regulski, I.: Writing habits as identity marker: on sign formation in Papyrus Gardiner II. In: Gülden, S., van der Moezel, K., Verhoeven, U. (eds.) Ägyptologische "Binsen"-Weisheiten III, pp. 235–265. Franz Steiner Verlag (2018). https://doi.org/10.25358/openscience-395
21. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000). https://doi.org/10.1126/science.290.5500.2323
22. Tabin, J.: From Papyrus to Pixels: Optical Character Recognition Applied to Ancient Egyptian Hieratic. B.A. research project, University of Chicago (2022). https://doi.org/10.6082/uchicago.3695
23. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000). https://doi.org/10.1126/science.290.5500.2319

24. Verhoeven, U.: Stand und Aufgaben der Erforschung des Hieratischen und der Kursivhieroglyphen. In: Verhoeven, U. (ed.) Ägyptologische "Binsen"-Weisheiten I–II, pp. 23–63. Franz Steiner Verlag (2013). https://doi.org/10.11588/propylaeumdok.00003321
25. Wang, Y., Yang, Y., Chen, H., Zheng, H., Chang, H.: End-to-end handwritten Chinese paragraph text recognition using residual attention networks. Intelligent Automation & Soft Computing **34**, 371–388 (2022). https://doi.org/10.32604/iasc.2022.027146