Encoding Hieroglyphic Texts

Heidi Jauhiainen[1]

[1]University of Helsinki, P.O. Box 4, 00014 University of Helsinki, Finland

Abstract

With the help of data science, researchers in the humanities can study large amounts of data at once and find regularities that they might not otherwise detect. In order to use digital methods, the texts to be examined must be in machine-readable form, but the lack of such text corpora hinders the digital study of ancient Egyptian texts. A sign can be next to, above, or over another in a hieroglyphic text, and two or more signs can be nested. Egyptologists use encoding to maintain the information on the signs and their places relative to each other when preparing hieroglyphic texts for publication in printed form. The encoding uses letter-number combinations from the Gardiner list, a standard reference list for Ancient Egyptian hieroglyphs. To increase the number of machine-readable hieroglyphic texts, the plan is to develop a workflow that uses automatic transliteration. This paper aims to present the first steps towards this goal. Ancient Egyptian texts are encoded by hand in JSesh, an open-source hieroglyphic editor. The aim is to publish annotated texts in a structured form, and a tool is being built to turn the binary format files produced in JSesh into machine-readable form. This paper introduces Gly2Mdc version 1.0, which extracts and cleans the encoding from the binary file. The tool is openly available and can be used for files with the extension .gly and containing encoded hieroglyphic text.

Keywords

Encoding, hieroglyphic texts, tool

1. Introduction

To use digital methods for researching texts, the texts must be machine-readable. For major modern languages, such as English, there are openly available digital corpora of texts which can be extensive and built from natively digital texts, such as Wikipedia. A smaller corpus is sufficient for many purposes, and there are several corpora built specifically for historical research. Assyriology, for example, has freely downloadable corpora of machine-readable cuneiform texts, such as Open Richly Annotated Cuneiform Corpus.[1] However, the lack of similar corpora hinders the digital study of ancient Egyptian texts.

The aim of the project *Machine-Readable Texts for Egyptologists*[2] is to develop a workflow for producing machine-readable hieroglyphic texts. OCRing hieroglyphic texts would produce machine-readable texts, but training the method would require a lot of annotated texts in the same handwriting; annotated texts that are currently not available. Using Unicode characters to build machine-readable hieroglyphic texts would also be an option, as there are over 1,000

The 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022), Uppsala, Sweden, March 15-18, 2022

heidi.jauhiainen@helsinki.fi (H. Jauhiainen)
0000-0002-8227-5627 (H. Jauhiainen)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] http://oracc.museum.upenn.edu

[2] https://blogs.helsinki.fi/hwikgren/marete/

hieroglyphic Unicode characters.[3] Unfortunately, the hieroglyphic block is outside the Basic Multilingual Plane, and the characters are not correctly handled by commonly used software applications such as Microsoft Word.

In a hieroglyphic text, the signs can be next to, above, or over another, or even nested. There is, therefore, a tradition of using encoding to maintain the information on the signs themselves and their places relative to each other when preparing hieroglyphic texts for printed publications. However, the encoded texts are usually discarded after the text has been built and turned into a picture. Egyptologists are, in fact, not used to handling hieroglyphic texts in encoded form. Instead, they transliterate hieroglyphic texts with Latin letters and diacritics as preparation for translating and analyzing them. Computer-assisted transliteration of hieroglyphic texts will speed up producing such texts in machine-readable form.

This paper presents the first stages of building an automated transliteration method for hieroglyphic texts and the first version of a tool to help in the process. To build the method, one needs machine-readable hieroglyphic texts. Since there is no working automated method for producing these, the chosen method is to encode texts by hand in a hieroglyphic editor called JSesh [1]. The encoded texts will be annotated with the automatically produced transliterations and eventually published in a structured form. For this end, a tool called Gly2Mdc[4] is being built. Version 1.0 of the Java-based tool handles the cleaning and writing the encoding from binary form to text file, a process that would otherwise be tediously slow.

I will first briefly describe the intricacies of hieroglyphic writing (Section 2) and why encoding is at the moment the best way of producing machine-readable hieroglyphic texts instead of, for example, Unicode fonts (Section 3). I will then outline the use of JSesh for producing encoded hieroglyphic texts (Section 4) before introducing version 1.0 of the Gly2Mdc tool for producing structured annotated files for publishing the machine-readable texts generated in the project (Section 5).

2. Hieroglyphic Text

Hieroglyphs were generally used in monumental texts. They could be written in different directions, from right-to-left, left-to-right, or in vertical columns continuing either right or left. The direction of a text usually depends on the related pictures. One can tell the writing direction from the living creatures amongst the hieroglyphs; one always approaches people and animals from the front.

When writing literature, letters, or administrative texts, ancient Egyptians generally used so-called hieratic writing, which is cursive handwriting that is based on hieroglyphic writing (see Figure 1). In the non-literary documents of the New Kingdom period (c. 1550-1069 BCE), hieratic was usually written from right to left.

Different hieroglyphic graphemes, generally called signs, have different functions, and one sign can have varying functions in different contexts [2]. Signs can be used as logograms, referring to entire words. Many signs function as phonograms representing one to three, rarely four, phonemes. Ancient Egyptian is an Afro-Asiatic language and, just as in Semitic languages,

[3] https://unicode.org/charts/PDF/U13000.pdf

[4] https://github.com/MaReTEgyptologists/gly2mdc

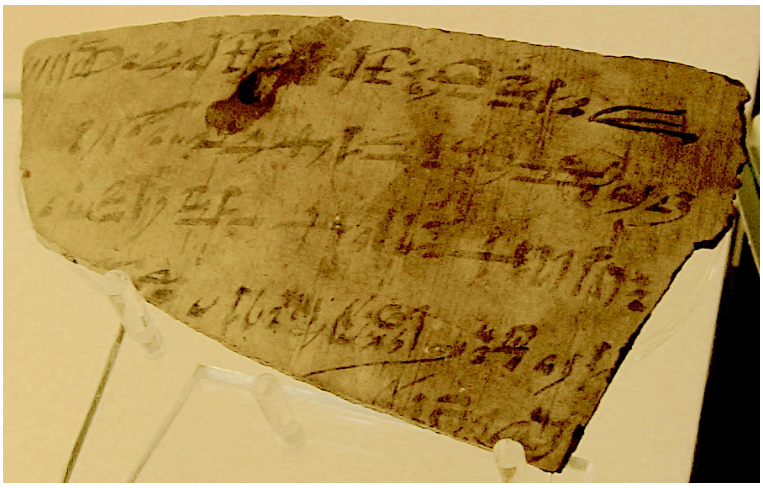
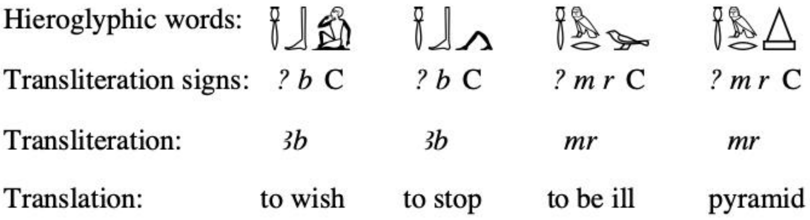


Figure 1: A hieroglyphic text written in hieratic handwriting on a piece of pottery. O. Turin N 57458, from Deir el-Medina, New Kingdom (c. 1550-1069 BCE). In Museo Egizio, Turin.

only consonants and semivowels were represented in writing. Both logograms and phonograms could be used to write words that had nothing to do with the sign they depict. Some signs have several different phonetic values, and several signs could have the same phonetic value. Therefore, signs were combined, and interpretants, or phonetic complements, were used to show which phonetic value was meant (see Figure 2). In order to tell the different words with the same phonetic values apart, classifiers were added to the end of the word.



Hieroglyphic words:	𓍑𓃀𓀁	𓍑𓃀𓂻	𓍑𓅓𓂋𓅪	𓍑𓅓𓂋𓉴
Transliteration signs:	*? b* C	*? b* C	*? m r* C	*? m r* C
Transliteration:	*ꜣb*	*ꜣb*	*mr*	*mr*
Translation:	to wish	to stop	to be ill	pyramid

Figure 2: The sign depicting a chisel that has two phonetic values *ab* (with Egyptological aleph) and *mr* is here used with phonetic complements and classifiers in different words. C stands for a classifier.

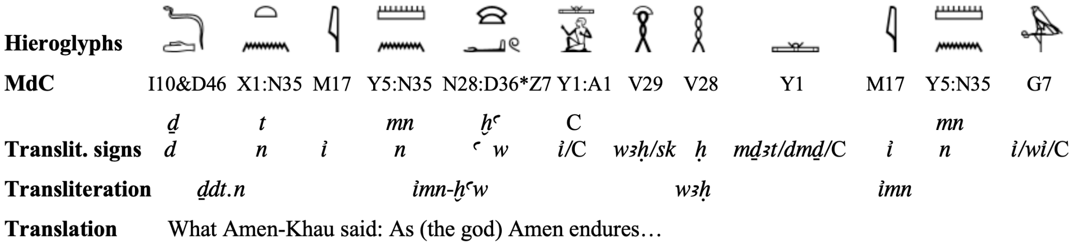
A Hieroglyphic text does not indicate word boundaries nor the end of a sentence. Furthermore, a hieroglyph was often written above or over another one, or they could be nested. Hieroglyphic words could also be written in multiple ways depending on the space and aesthetic preferences of the scribe. Producing machine-readable hieroglyphic texts is, hence, not straightforward.

3. Machine-Readable Hieroglyphic Texts

One way of producing machine-readable hieroglyphic texts would be to use Unicode, which, since 2009, includes over 1000 hieroglyphic signs. In 2019, the so-called format control characters for positioning the hieroglyphic signs were introduced in Unicode version 12, and, since then, it has been possible to position the signs properly. However, writing hieroglyphic texts with Unicode code points—for example, 13001 for the sitting man in Figure 2—is not easy. There are fonts for Unicode hieroglyphs, but they require specific tools depending on whether one wants to use them, for example, on a web page or a document. One cannot just download the font and start writing hieroglyphs in a text file. Moreover, none of the available hieroglyphic keyboards seem to provide the possibility of positioning the signs using the newly introduced format control characters.[5]

Transliteration refers to the conversion of text written using one writing system to another, and Egyptologists use it to transform texts written in hieroglyphs into words written in Latin letters and diacritical marks. The various transliteration fonts for ancient Egyptian texts are far easier to use than Unicode characters for Egyptian hieroglyphs. Hence, producing transliterated hieroglyphic texts in machine-readable form would be possible. However, the conversion is more complex than in many other languages, as the writing system uses characters that correspond to more than one Latin letter or have alternatives for equivalences. Because many hieroglyphs, particularly the groups formed by them, can be transliterated in many ways, manual transliteration of a hieroglyphic text requires examination of vocabularies and conclusions drawn from that, making it a slow process. Transliteration is already an interpretation of the text, and, more importantly, it does not retain the information on the hieroglyphic signs used.

In order to display the signs properly in printed books and online texts, the hieroglyphic texts are encoded with Latin letters and Arabic numbers (see Figure 3). The codes come from the so-called *Gardiner Sign List*,[6] the standard reference list for Egyptian hieroglyphs. The list was compiled by Sir Alan Gardiner in 1927 for his grammar of the Middle Egyptian language [3], and the letters refer to various categories of signs while each sign has a number within its category.



Hieroglyphs												
MdC	I10&D46	X1:N35	M17	Y5:N35	N28:D36*Z7	Y1:A1	V29	V28	Y1	M17	Y5:N35	G7
Translit. signs	*ḏ* *d*	*t* *n*	*ỉ*	*mn* *n*	*ḫʿ* *ʿ w*	C *ỉ*/C	*wꜣḥ/sk*	*ḥ*	*mḏꜣt/dmḏ*/C	*ỉ*	*mn* *n*	*ỉ/wỉ*/C
Transliteration	*ḏdt.n*			*ỉmn-ḫʿw*			*wꜣḥ*			*ỉmn*		
Translation	What Amen-Khau said: As (the god) Amen endures…											

Figure 3: The first words of the text in Figure 1 transcribed from the cursive original to hieroglyphs with Manuel de Codage (MdC) encoding. The line with the transliteration of the signs displays the alternatives when available. C stands for a classifier. A person with the name Amen-Khau is swearing an oath.

There are a few different methods of encoding hieroglyphs [4], the most used of which is called Manuel de Codage (MdC) [5]. To place a sign above another, MdC uses a colon, and an asterisk is used to indicate that signs are next to each other [6]. However, MdC does not indicate different sizes of signs, and there is no way of placing a sign over another or nesting them. MdC is used in various hieroglyphic editors that are often operating system specific, and

[5] E.g. Keyman Hieroglyphic keyboard https://help.keyman.com/keyboard/hieroglyphic/1.4/hieroglyphic.

[6] https://en.wikipedia.org/wiki/Gardiner's_sign_list

each software has dealt with these shortcomings differently [7].

Mark-Jan Nederhof proposed a new encoding system which he called Revised Encoding Scheme (RES) [5]. With RES, signs can be nested and positioned over each other in multiple ways. The position is not absolute; instead, it is defined in relation to other signs in the group. RES has not been widely adopted by Egyptologists [8]. Because of its precise and lengthy commands, they find RES too slow to write. It is also considered unnecessary since Egyptologists are accustomed to the graphical hieroglyphic editors using MdC.

The encoded hieroglyphic texts are machine-readable, and Egyptology has, thus, a tradition of producing machine-readable texts. Unfortunately, the encoding has only been considered an intermediate step when publishing the texts as pictures in books and articles. There is no tradition of publishing the encoding of the texts, and, instead, these are often discarded [7].

4. Encoding with JSesh

In the *Machine-Readable Texts for Egyptologists* project, I have chosen to encode hieroglyphic texts with MdC using the software called JSesh [1]. JSesh is an open-source, Java-based, operating system independent word processor for producing hieroglyphic texts. In order to deal with the shortcomings of MdC, JSesh uses some additions to the encoding scheme. For example, one can use the ’##’-code to place a sign over another. ’&’ and ’^^^’ can be used to group and nest signs, respectively. There are also prefabricated composite signs, and it is possible to position and resize signs manually. The texts are saved as binary files (.gly) compatible with other hieroglyphic editors, such as Winglyph and Tksesh. The texts produced can be exported as pictures, PDFs, or in RTF format, and it is possible to copy out the MdC encoding or the text as Unicode characters.

Since the aim is to produce machine-readable texts, not pictures, the attention is on the MdC encoding. The texts will be published for future use in a structured and annotated format using TEI markup language,[7] the structure preferred in Egyptology [9]. In order to be able to build the TEI formatted files correctly, it is essential that the MdC encoding is as clean as possible. Resizing and positioning signs manually in JSesh clutters the encoding. For example, it is possible to nest sign D46 inside sign I9 by hand, but the encoding will end up looking like this: I10\81**D46{{36,525,63}} while I10&D46 would require less complicated rules when analyzing the encoded texts digitally. Thence, no manual positioning and resizing will be done to the texts.

In the future, operating hieroglyphic Unicode characters might be more straightforward and, hence, it is considered beneficial to annotate the texts with Unicode characters in addition to the transliteration. Therefore, care is taken not to use signs or codes that are not in the Unicode chart for hieroglyphs,[8] although additional sign-code combinations are available in JSesh.

[7] Text Encoding Initiative, https://tei-c.org

[8] https://www.unicode.org/charts/PDF/U13000.pdf

5. Gly2Mdc

In JSesh, there is no option of saving the encoding directly to a text file; instead, it has to be copied by hand to a new file. This operation requires many steps even when the chosen default copy mode is the encoding: 1. select all the text, 2. copy, 3. open a new file in a text editor, 4. paste, and 5. save the new file under a chosen name. To produce the TEI formatted files that contain the annotated MdC encoding, it is thus, best to export the encoding directly from the binary files with the extension .gly. For this purpose, I have built a tool called Gly2Mdc.

Gly2Mdc version 1.0 extracts the MdC encoding from the binary file, cleans the text, and writes it to a new text file. The binary file contains information on the settings used when producing the hieroglyphic text in the editor. These extra lines are removed during the cleaning process. In the MdC encoding, a code is separated from other codes with a hyphen unless the signs are marked as forming a sign combination. In JSesh, the codes are separated with an underscore-hyphen combination (e.g., O1_-D21:X1*N5_-A24). The underline and hyphen are replaced with a single white space during the cleaning to make the reading and future processing of the encoding easier.

Gly2Mdc has been built using Java, and both an executable jar file and the source code are openly available on GitHub with some sample texts.[9] The tool takes the binary file produced with JSesh as input and produces the encoded text to a file with the same name but extension .gly2mdc. It works with a simple command `java -jar Gly2mdc.jar <binary_file_name.gly>`, where 'binary_file_name' is the name of the file containing the text produced with JSesh. In the future versions of Gly2Mdc, the tool will be expanded to writing the annotated text using TEI markup language.

6. Conclusions

Gly2Mdc can be used by anyone producing hieroglyphic texts with JSesh and other hieroglyphic editors that save the files in binary format with the extension .gly. Instead of being thrown away after the texts has been saved as a pictures or PDFs, the binary files can be turned into text files with the encoded texts in machine-readable form for others to use.

Since JSesh version 7.5.5 was released in 2020, it is possible to copy the text in Unicode characters, but that takes as many steps as copying the MdC. More, in fact, since only MdC can be set as the preferred clipboard format and copying Unicode requires going through the Edit menu and choosing which format to copy. In order to bypass this tedious process, Gly2Mdc will be expanded to add the Unicode character to the metadata of each sign. A possibility of writing the text to a text file in Unicode hieroglyphs is also planned.

Acknowledgments

I gratefully acknowledge the funding received from the Finnish Cultural Foundation and the Kone Foundation that has made this research possible. Furthermore, I would like to extend my gratitude to Dr. Tommi Jauhiainen for reading and commenting on the draft of the article. I

[9] https://github.com/MaReTEgyptologists/gly2mdc

am also grateful to assistant professor Saana Svärd for including me in the inspiring research collective of the Academy of Finland Center of Excellence in Ancient Near Eastern Empires. Finally, I would like to thank the anonymous reviewers for their valuable comments.

References

[1] S. Rosmorduc, JSesh Documentation. [online], 2014. URL: http://jseshdoc.qenherkhopeshef.org.

[2] S. Polis, The Functions and Toposyntax of Ancient Egyptian Hieroglyphs: Exploring the Iconicity and Spatiality of Pictorian Graphemes, Signata: Annales des Sémiotiques 9 (2018). doi:10.4000/signata.1920.

[3] Sir A. Gardiner, Egyptian Grammmar: Being an Introduction to the Study of Hieroglyphs, 3rd. ed., Griffith Institute, Oxford, 1957.

[4] R. B. Gozzoli, Hieroglyphic Text Processors, Manuel de Codage, Unicode, and Lexicography, in: S. Polis, J. Winand (Eds.), Texts, Languages Information Technology in Egyptology, Presses Universitaires de Liège, Liège, 2013, pp. 89–101.

[5] M.-J. Nederhof, A Revised Encoding Scheme for Hieroglyphic, in: Proceedings of the XIV Computer-aided Egyptology Round Table, Pisa, Italy, July 2002, IE2002, 2002.

[6] J. Buurman, N. Grimal, M. Hainsworth, J. Hallof, D. van der Plas, Inventaire des signes hiéroglyphiques en vues de leur saisie informatique: Manuel de codage des textes hiéroglyphiques en vue de leur saisie sur ordinateur, Institut de France, 1988.

[7] M.-J. Nederhof, The Manuel de Codage Encoding of Hieroglyphs Impedes Development of Corpora, in: S. Polis, J. Winand (Eds.), Texts, Languages Information Technology in Egyptology, Presses Universitaires de Liège, Liège, 2013, pp. 103–110.

[8] M.-J. Nederhof, Automatic Alignment Of Hieroglyphs And Transliteration, Gorgias Press, 2009, pp. 71–92. URL: https://doi.org/10.31826/9781463216269-007. doi:doi:10.31826/9781463216269-007.

[9] D. A. Werning, Towards Guidelines for TEI Encoding of Text Artefacts in Egyptology: TEI Templates, Thesauri, ”EpiDoc” cheatsheet, 2016. URL: https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa-221617.