

TIR over Egyptian Hieroglyphs

Estíbaliz Iglesias-Franjo, Jesús Vilares
Grupo Lengua Y Sociedad de la Información (LYS), Departamento de Computación
Facultade de Informática, Universidade da Coruña
Campus de A Coruña, 15071 – A Coruña (Spain)
Email: estibaliz.ifranjo@udc.es, jesus.vilares@udc.es
Website: <http://www.grupolys.org>

Abstract—This work presents an Information Retrieval system specifically designed to manage Ancient Egyptian hieroglyphic texts taking into account their peculiarities at both the lexical and encoding level for its application in Egyptology and Digital Heritage. The tool has been made freely available to the research community under a free license and, to the best of our knowledge, is the first tool of its kind.

Keywords—Archaeology; Digital Heritage; Egyptology; hieroglyphs; manuel de codage; Text Retrieval

I. INTRODUCTION

In recent decades the research community has focused its efforts on developing Text Mining systems, including Information Retrieval (IR) systems, for contemporary languages, which from a socio-economic point of view, makes a great deal of sense. However, in recent years we have seen a growing interest in the field of *Digital Humanities*, the science area which deals with the application of computing technologies to the various disciplines of Humanities: from Philosophy to Linguistics and from History to Music. One of its branches is the so-called *Digital Heritage*, which focuses on the use of computing and information technologies for the preservation and study of our cultural legacy.

One of the fields which may benefit from these technologies is Egyptology. This paper describes an open source Text Information Retrieval (TIR) system designed specifically to work with the Egyptian hieroglyphic writing system. To the best of our knowledge this is the first tool of its kind.

The rest of this work is structured as follows. Firstly, Section II sets the context by presenting the main features of the Egyptian writing system and Section III explains how to encode hieroglyphic texts. Next, Section IV introduces the main text processing tools used in Egyptology. Section V analyses the requirements of the system while Section VI presents its architecture. Finally, Section VII presents our contributions and ideas for future work.

II. THE EGYPTIAN LANGUAGE

A. History

Egyptian is the longest-attested human language, with a documented history that spans from around 3300 BC until the present day, when it continues to be used by the

Coptic Christian Church in its rituals. Egyptian language has undergone very deep changes at all levels throughout its lifetime [1], [2]. Because of its archaeological interest, our work focuses on the so-called *Middle Egyptian* or *Classic Egyptian*, which corresponds to the stereotypical image we have of Egyptian. It was spoken from around 2100 BC until 600 BC, but remained as the traditional language of hieroglyphic inscriptions until the fifth century AD, thus still being widely used in royal inscriptions, religious literature and monuments.

B. Characteristics of the Language

Egyptian is an Afro-Asiatic language, as are Arabic and Hebrew, but constitutes a subfamily of its own. Regarding its writing system, we should single out these features [2], [1]:

- *Pictographic*. Its signs for writing, known as *hieroglyphs*, consist of symbols portraying elements of their world: parts of the body (𓂀), animals (𓂁), objects (𓂂), etc.
- *Logographic*. Some of the symbols have a meaning that corresponds, directly or indirectly, to the same real-world element they reproduce. For example: an *eye* (𓂃) for “eye” or a *mast with sail* (𓂄) for “wind”.
- *Phonographic*. Signs may represent sounds. However, Egyptian was a *consonantal* language where only the consonants of the word were written, as in the case of early Arabic and Hebrew. For example: 𓂀 corresponds to the phoneme /b/, transliterated as *b*.¹

So, Egyptian combined several types of signs to form words and phrases [1], [2]:

- *Phonograms*. These signs were used by convention to represent the sounds of language. We can distinguish three types according to the number of consonantal sounds represented: *unilateral* (e.g. 𓂀, *b*);² *biliteral* (e.g. 𓂁, *sb*); and *triliteral* (e.g. 𓂂, *bpr*).

¹Transliteration consists of representing the signs of a given writing system with those of another.

²Where appropriate we will indicate, as in this case, the transliteration corresponding to the hieroglyphic text in question.

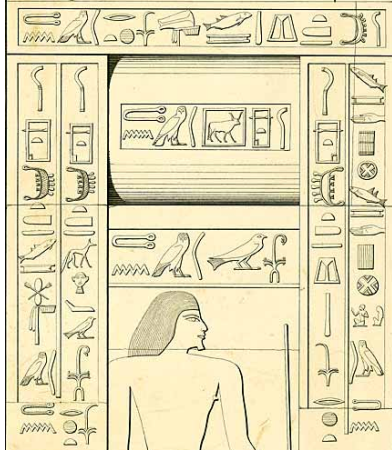








Figure 1. Part of the false door found in the tomb of a high official. The upper half shows text written in rows (to be read right-to-left) while the laterals contain text written in columns (to be read right-to-left in the case of the left side, and left-to-right in the case of the right side).







- *Ideograms* (aka *logograms*). They represent the things they actually depict and, consequently, are read that way. For example , that depicts a scribe's kit and is read *sbj*, is used for "write" and related words.
- *Determinatives*. These signs indicate that the word corresponds to a given semantic group, thus allowing the reader to differentiate between words having the same consonantal representation but a different meaning. Determinatives are silent so they are not read. As an example, given the previous ideogram  and the determinatives  and , the word  means "to write" while the word  means "scribe".

It should be noted that the same sign may belong to more than one of these categories at the same time.

C. Writing Direction

Another feature of Egyptian is its flexibility with regard to its *writing direction*. Hieroglyphic texts can be found written in horizontal rows, as with English, or in vertical columns, as with traditional Japanese and Chinese. Moreover, although they are always read from top to bottom, they may follow a left-to-right ordering, as with English, or a right-to-left ordering, as with Arabic and Japanese. This is due to the fact that Egyptian writing had a marked artistic nature [2], [1] since it was intended to be carved or painted on monuments, walls, statues, etc. Since one of the main characteristics of Ancient Egyptian art was its symmetry, Egyptian writing necessarily had to conform to this requirement. Figure 1 shows a good example of this.

D. Sign Groups




Egyptian was written continuously, as in the case of Chinese or Japanese, with no word or phrase delimiters. Additionally, hieroglyphs were not arranged one after the other, in a linear way, as in the case of our alphabetic system. Instead, scribes gathered them together in so-called *groups* [1] in a way which resembles contemporary Hangul Korean script. For example, the word "sycamore" (*nht*) was not written    but    instead. This was done following a series of principles or heuristics [2] seeking to obtain the most harmonious and aesthetic arrangement:

- *Symmetry*. Small and horizontal signs will be written centered within the group if they appear alone.
- *Horror vacui*. Abhorrence of empty space.
- *Minimization*. If necessary, large signs may be partly reduced to group them with other symbols.

III. HOW TO ENCODE HIEROGLYPHIC TEXTS


Contemporary scholars needed a practical way to represent hieroglyphs without re-drawing them. The solution consisted of encoding the signs using regular characters.





A. Gardiner's List

In the so-called *Gardiner's List* [3], a standard reference in the study of Egyptian, signs are classified into 26 categories according to their drawing, each one identified with a letter: A corresponds to "Man and his occupations" (...); B to "Woman and her occupations" (...); etc. In turn, hieroglyphs within each category are numbered sequentially. Thus, a given sign can be coded using the letter of its category and its number within the group; e.g. G5 corresponds to sign  ("falcon"), the fifth element of category G ("Birds").

B. Manuel de Codage





In the 80s, the *International Association of Egyptologists (IAE)*³ formed a committee with the aim of developing a standard encoding system for the digitalization of hieroglyphic texts. The resulting document was the *Manuel de Codage (MdC)* [4], an evolution of *Gardiner's List* where new codes and rules had been added for the accurate representation of hieroglyphs and other features of the language by using ASCII text. Next, we will introduce an overview of the most significant additions.



1) *Sign Arrangement*: Table I shows, in order of precedence, the basic operators for arranging the signs. Thus, 

    ("The birds are on the sycamore") is M17-G43-G1-Q3:D46-G43-G38:Z2-D2:Z1-N35:O4*X1-M1.

³<http://www.iae-egyptology.org/>

Table I
SIGN ARRANGEMENT OPERATORS IN MdC.

Symbol	Operation	Example
-	concatenation	Q3-X1-Z4-N1 
:	subordination	X1:Z4:N1 
*	juxtaposition	Q3*X1:Z4 
()	grouping	Q3*(X1:Z4):N1 

2) *Damaged Texts*: One of the specific problems to be faced in this context was the representation of damaged texts in the most informative way. This matter was solved by the use of *shades*, implemented as marks attached to the sign codes, allowing us to express whether the sign or even its presence is recognizable or not, how many signs are affected, which parts of them are damaged, etc. For instance, given our previous example , codified as M17-G43-G1-Q3:D46-G43-G38:Z2-D2:Z1-N35:O4*X1-M1, if we suppose that the entire second symbol is blurred but recognizable, the third sign has completely disappeared and the upper part of the last one is damaged, it would be codified M17-G43#-//Q3:D46-G43-G38:Z2-D2:Z1-N35:O4*X1-M1#12 instead, and its corresponding graphical representation would be .

3) *Non-Hieroglyphic Text*: MdC includes encoding support for combining hieroglyphs, transliterations, translations and other types of annotation within the same text. It assumes that all text is hieroglyphic unless it is enclosed between the marks '+t' (opening) and '+s' (closing) in the case of transliterations, or '+l' (opening) and '+s' (closing) in the case of regular Latin text.

IV. HIEROGLYPHIC TEXT PROCESSING

In Egyptology, computer processing of hieroglyphic text has been closely linked to the development of classic-style text editors [5], [6]. Since there were no hieroglyphic typewriters, scholars relied on handwritten texts when writing and sharing documents. Even in the case of books, the hieroglyphic texts printed in their pages were typographical transcriptions or, more commonly, mere copies of those handwritten by their authors.

Among the specialized, and also scarce, text processor software developed for this purpose, we should highlight two tools. Firstly, the word processor GLYPH [5], developed by Jan Buurman, one of the designers of the MdC. This initial tool, which laid the foundations of future hieroglyphic text processors, was published for DOS in 1986 and subsequently evolved and migrated to other operating systems. The second

tool we want to cite is JSESH [7], developed by Serge Rosmorduc, which is, currently and in all probability, the most widely used word processor in Egyptology.

V. REQUIREMENTS OF THE SYSTEM

In this project our goal has been the development of a TIR system capable of operating on Egyptian texts. After consulting an expert Egyptologist and studying the nature of the language, we extracted its initial requirements from the point of view of its potential future users:

- 1) **Simplicity**: It should be intuitive and easy to use.
- 2) **Content indexing**: The system must be able to index documents containing conventional text and hieroglyphic text. We will initially focus on those documents written with JSESH, thus covering a large proportion of the digitalized contents currently available.
- 3) **Querying using MdC encoding**: In the case of hieroglyphs, the user will input the query using MdC encoding, with which he or she is already familiarized.
- 4) **Display the query using glyphs**: In order to make it easier for the user, the system will display, in parallel, the input MdC query using pictograms.
- 5) **Querying using Latin text**: Since the documents contain both hieroglyphic and conventional Latin text, we also want to be able to submit conventional text queries.
- 6) **Submission of mixed queries**: Possibility of making queries combining both hieroglyphic and Latin text.
- 7) **Relevant documents retrieval**.
- 8) **Display of document contents**: The user should be able to access the content of the documents retrieved by the system and check why they have been retrieved.

On the basis of these requirements, we have developed our TIR system, now publicly available at:

<http://github.com/estibalizifranjo/hieroglyphs>

VI. ARCHITECTURE OF THE SYSTEM

Figure 2 shows a schematic representation of our system. We can distinguish two main phases: firstly, the indexing of the document collection on which searches are to be performed and, secondly, the querying-retrieval process.

A. Phase 1: Indexing

During this stage the system extracts the contents of the input documents, separating the Latin text and the hieroglyphic text. Each kind of text will be processed separately for later indexing. Next, we describe the different processes involved.

1) *Content Extraction*: Our engine works on text documents, but input documents may be .PDF, .DOC, .ODT, etc. So, this first module uses the Apache Tika toolkit⁴ to detect and extract the text contained in those input documents.

⁴<http://tika.apache.org>

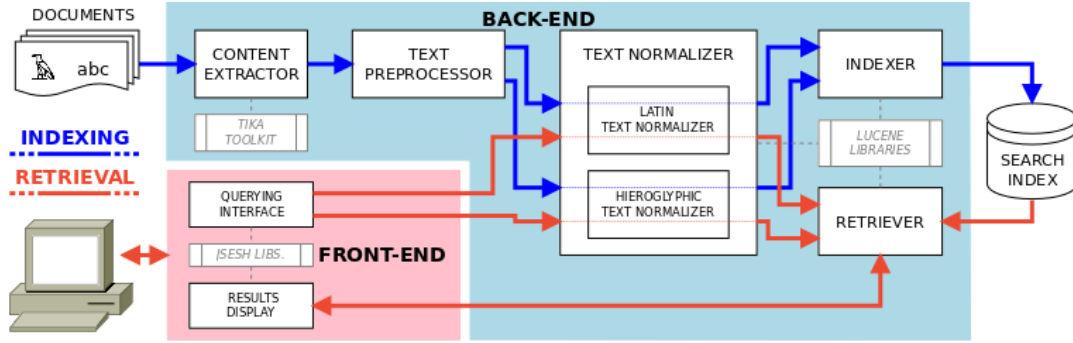


Figure 2. Architecture of the system: indexing (blue) and retrieval (reddish) processes.

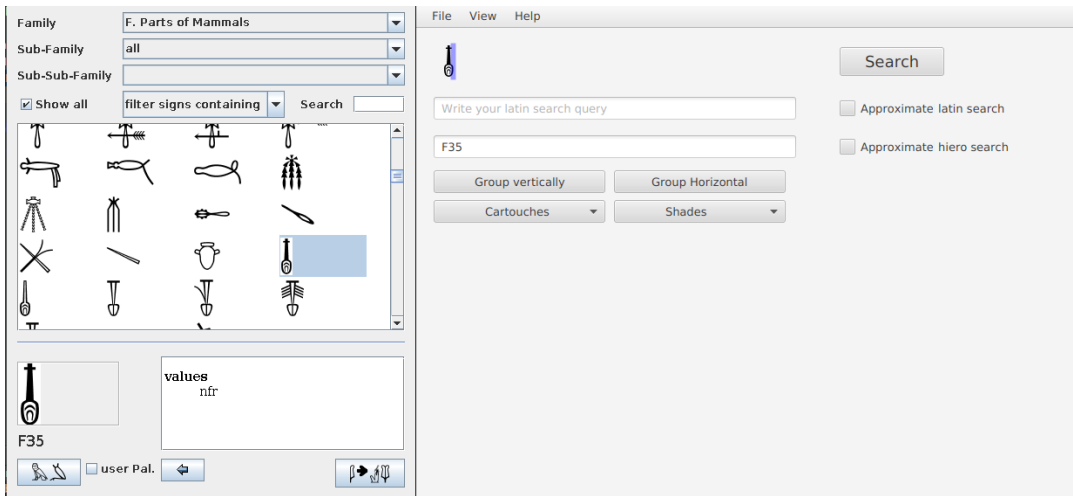


Figure 3. Querying interface.

2) *Text Preprocessing*: That content is then preprocessed using *pattern matching* in order to separate conventional text from hieroglyphic text and to filter out useless data.

3) *Latin Text Normalization*: The normalization components apply a series of *text operations* for tokenizing, conflating and generating the index terms of the input texts. The nature of such operations varies according to the type of text: Latin text or hieroglyphs. For its implementation we have taken as our basis Apache Lucene.⁵ In the case of Latin text, a standard processing is performed [8]: firstly, a standard lexical analysis is applied for tokenizing the text, and the resulting terms are then conflated by lowercasing them and removing both stopwords and diacritics.

4) *Hieroglyphic Text Normalization*: In this case, the processing is completely different and presents several complications due to the peculiarities of this writing system. The first problem is its continuous writing, with no delimiters to separate words or phrases. In this first distribution of our system we have opted to use *sign groups* (see Section II-D) as our working unit since, in the case of the codified

document, they are delimited by '-'. For example, the word $\overline{\square\Delta}\overline{\cup}$ (N35:04*X1-M1) is composed of four signs but only two groups, so it would be tokenized into $\overline{\square\Delta}$ (N35:04*X1) and $\overline{\cup}$ (M1). Additionally, unlike the Latin text processor, input text will not be lowercased, since MdC encoding is case-sensitive. This is also the case of punctuation marks, which form part of MdC encoding.

5) *Index Generation*: Finally, an index structure is generated taking as input the index terms obtained from the documents. In the case of the hieroglyphic text, the sign groups are indexed together with their occurrence positions within the text. This module has been also implemented using Lucene.

B. Phase 2: Querying and Retrieval

Two main sub-processes can be distinguished in this second phase, namely the querying process and the retrieval process:

1) *Querying*: Our requirements state that the user must be able to query the indexed collection by using either hieroglyphics, Latin text or a combination of both (*mixed*

⁵<http://lucene.apache.org/core/>

queries). The query normalization process is parallel to that performed during the indexing. In the case of hieroglyphic text, two search modes are available at this time: *exact matching*, where we require the documents to contain exactly the same group sequence specified in the query (i.e. the same signs with the same arrangement); and *approximate matching*, which allows the user to sub-specify the composition of a group (e.g. to require that a given group of the sequence contains a given sign but without specifying whether it contains any more symbols or their arrangement within the group).

2) *Retrieval*: The recovery module searches the index and identifies those documents that are relevant to the query. The resulting list of documents is returned to the user.

3) *Front-End Interface*: The interface of the system has been designed to make its use as easy and intuitive as possible. Thus, the user is provided with separate search forms for Latin and hieroglyphic queries, as can be seen in the right-hand panel of Figure 3. In the case of hieroglyphic queries, those pictograms corresponding to the MdC code text being introduced will be displayed so that the user can check them on the fly. Moreover, we have integrated extra features which provide the interface with improved flexibility while greatly simplifying its use. Thus, the interface provides the user, if required, with a palette of hieroglyphic signs that can be added to the query simply by clicking on them, as shown in the left-hand panel of Figure 3. The palette also functions as a catalog of symbols organized according to Gardiner’s List classification. Furthermore, the interface provides several options for handling the hieroglyphic text, such as adding shadows or creating your own palettes. It is also responsible for presenting the user with the result of the search and for accessing the content of the documents, which, if required by the user, will be displayed highlighting the matchings found. Finally, the interface supports *internationalization*. At this time the user can choose between English, French, Spanish and Galician.

For its implementation we have used the libraries provided with the JSesh editing tool [7], including its symbol palette. This way, users of the system will find an interface with a very similar appearance and behavior to that of the tool they are already familiar with, thus minimizing the learning curve.

VII. CONCLUSIONS AND FUTURE WORK

To the best of our knowledge, this work presents the first tool of its kind, a Text Information Retrieval system designed to work with Egyptian hieroglyphic texts, taking into account their peculiarities at both the lexical and encoding level. The system admits queries containing both hieroglyphic and Latin text, and its user interface has been designed to make it as intuitive and easy to use as possible. This tool has been released under a free license for use by the research community.

With respect to future work and from an applicative point of view, new input filters would allow the system to accept as input source new types of documents containing hieroglyphic text, such as Unicode text documents [9] or, as in the case of this article, L^AT_EX documents built using the package HieroT_EX [10]. Moreover, our system has been initially configured to use a Boolean model [8] for relevant document selection, which is based on exact group-sequence matching with the possibility of sub-specifying the composition of the group. Those documents are then re-ranked by using a Vector Space model. However, we would like to try more flexible approaches, such as using a pure Vector Space model solution, always taking into account the needs of the intended users.

However, from an academic point of view, the work should focus on dealing with the complex nature of this language and its context. After analyzing the case of other languages which share characteristics with Egyptian, such as Chinese [11], Korean [12] or Arabic [13], we believe that the use of sign or group *n-gram based processing* could deal with the problems deriving from its continuous writing and the noisy character of the texts, either because of the redundancies derived from the common use of phonetic complements⁶ or the presence of deteriorated texts. In fact, we are currently working on a preliminary implementation of a solution based on group *n-grams*. Other possibilities to be considered are the use of *phonetic matching* [14] or *conflation mechanisms* based on lemmatization or morphological analysis [15]. All these solutions would require a further study of the language and the development of resources such as evaluation corpora, which were beyond the scope of this initial project. However, we intend to contact experts in the field and initiate the development of a small corpus for basic evaluation tasks. This corpus would also be made available to the research community.

ACKNOWLEDGMENT

Research partially funded by the Spanish Ministry of Economy and Competitiveness–MINECO (project FFI2014-51978-C2-2-R). We would like to thank Dr. Josep Cervelló Autuori, Director of the Institut d’Estudis del Pròxim Orient Antic (IEPOA) of the Universitat Autònoma de Barcelona, for introducing us to Egyptian; and Dr. Serge Rosmorduc, Associate of the Conservatoire National des Arts et Métiers (CNAM), for his support with JSesh.

REFERENCES

- [1] J. P. Allen, *Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs*, 3rd ed. Cambridge, UK: Cambridge University Press, 2014.

⁶In Egyptian, bi/trilaterals used to be written together with *phonetic complements* [1], [2], shorter phonograms (uni/bilaterals) that “spell out” part (if not all) the sounds encoded in the bi/trilateral.

- [2] J. Cervelló-Autuori, *Escrituras, Lengua y Cultura en el Antiguo Egipto*, ser. El espejo y la lámpara. Bellaterra: Edicions UAB, 2015.
- [3] A. H. Gardiner, *Egyptian grammar: being an introduction to the study of hieroglyphs*, 3rd ed. Oxford: Griffith Institute, Ashmolean Museum.
- [4] J. Buurman, N.-C. Grimal, M. Hainsworth, J. Hallof, and D. van der Plas, *Inventaire des signes hiéroglyphiques en vue de leur saisie informatique: manuel de codage des textes hiéroglyphiques en vue de leur saisie sur ordinateur*, ser. Mémoires de l'Académie des Inscriptions et Belles-Lettres. Paris: De Boccard, 1988, vol. 8.
- [5] R. Gozzoli, *Texts, Languages & Information Technology in Egyptology*, ser. Collection Ægyptiaca Leodiensia. Liège: Presses Universitaires de Liège, 2013, vol. 9, ch. Hieroglyphic Text Processors, Manuel de Codage, Unicode and Lexicography, pp. 89–101.
- [6] N. Grimal, “Hiéroglyphes et ordinateurs,” *BRISES. Bulletin de Recherches sur l'Information en Sciences Économiques Humaines et Sociales*, no. 15, pp. 57–60, 1990.
- [7] S. Rosmorduc, “JSESH documentation,” 2014. Available at: <http://jsesh.qenherkhopeshef.org/> (visited on June 2016).
- [8] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, USA: Cambridge University Press, 2008.
- [9] U. Consortium, “The Unicode standard, version 8.0. Egyptian hieroglyphs: Range 13000–1342f.” 2015. Available at: <http://www.unicode.org/charts/PDF/U13000.pdf> (visited on June 2016).
- [10] S. Rosmorduc, “HieroTeX: A LaTeXperiment of hieroglyphic typesetting,” 2003. Available at: <http://www.ctan.org/tex-archive/language/hieroglyph> (visited on June 2016).
- [11] S. Foo and H. Li, “Chinese word segmentation and its effect on information retrieval,” *Information Processing and Management*, vol. 40, no. 1, pp. 161–190, 2004.
- [12] J. H. Lee and J. S. Ahn, “Using n-grams for Korean text retrieval,” in *Proc. of 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*. New York, USA: ACM, 1996, pp. 216–224.
- [13] S. H. Mustafa and Q. A. Al-Radaideh, “Using n-grams for Arabic text searching,” *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 55, no. 11, pp. 1002–1007, 2004.
- [14] M. Yasukawa, J. S. Culpepper, and F. Scholer, “Phonetic matching in Japanese,” in *Proc. of SIGIR 2012 Workshop on Open Source Information Retrieval (OSIR 2012)*, 2012, pp. 68–71.
- [15] M. Piotrowski, *Natural Language Processing for Historical Texts*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.