

Any papyrus about “a hand over a stool and a bread loaf, followed by a boat”? Dealing with hieroglyphic texts in IR

Estíbaliz Iglesias-Franjo
estibaliz.ifranjo@udc.es

Jesús Vilares
jesus.vilares@udc.es

Grupo LYS, Departamento de Computación
Fac. de Informática, Universidade da Coruña
Campus de A Coruña, 15071 – A Coruña (Spain)

ABSTRACT

Digital Heritage deals with the use of computing and information technologies for the preservation and study of the human cultural legacy. Within this context, we present here a Text Retrieval system developed specifically to work with Egyptian hieroglyphic texts for its use by Egyptologists and Linguists in the study and preservation of Ancient Egyptian scripts. We intend to make it freely available to the Egyptology research community. To the best of our knowledge this is the first tool of its kind.

CCS Concepts

•Information systems → Document representation; Search interfaces; Multilingual and cross-lingual retrieval; Digital libraries and archives; •Applied computing → Archaeology; Arts and humanities;

Keywords

Text retrieval; digital heritage; egyptology; egyptian hieroglyphs; *manuel de codage*

1. INTRODUCTION

In recent years we have seen a growing interest of the research community in the fields of Digital Humanities and Digital Heritage. In a broad sense, *Digital Humanities*, also known as *Humanities Computing*, is the science area which deals with the application of computing technologies to the various disciplines of Humanities and closely related Social Sciences: from Philosophy to Linguistics and from History to Music [10]. 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 the human cultural legacy. As stated by the UNESCO, it embraces cultural, educational, scientific and administrative resources, as well as technical, legal, medical and other kinds of information created digitally, or converted into digital form

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CERI 2016 June 14–16, 2016, Granada, Spain

© 2016 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

from existing analogue resources. Many of these resources have lasting value and significance, and therefore constitute a heritage that should be protected and preserved for current and future generations [17].

Our work focuses on Egyptology and, more specifically, the study and preservation of Ancient Egyptian hieroglyphic scripts. This contribution describes an open source Information Retrieval (IR) system designed specifically to work with the hieroglyphic writing system used in Ancient Egypt. Thus, our system can deal with document collections containing not only regular text, but also digitalized hieroglyphic texts. In the same way, users are also enabled to submit queries using hieroglyphs when searching hieroglyphic documents about a given topic. To the best of our knowledge this is the first tool of its kind.

The structure of the rest of this work is as follows. Firstly, Section 2 sets the context by describing the main features of Ancient Egyptian writing. Next, Section 3 presents how these hieroglyphic texts can be encoded for their management. The requirements of our system are analysed in Section 4, while the tool itself is described in Section 5. Section 6 provides a brief overview of related work, focusing on those software tools used for editing hieroglyphic documents. Finally, Section 7 presents our conclusions and proposals for future developments.











2. ANCIENT EGYPTIAN WRITING

2.1 Nature of the Language

The Egyptian hieroglyphic writing system [3, 1] emerged in the region of Egypt around the year 3300 B.C. and its use lasted until the fourth century A.D. It is characterised by being:

- *pictographic*, since its symbols portray beings and objects of their reality: parts of the human body, plants, animals, scenes of everyday life, etc.;
- *logographic*, since it has symbols whose meaning correspond to the same real-word object they reproduce;
- *phonographic*, since some hieroglyphics depict sounds;
- *consonantal*, only the consonants are represented (as in the case of early Arabic and Hebrew scripts).

Moreover, not all hieroglyphs will be interpreted in the same way or will have the same functions, there thus being three categories of signs:


1. *Phonograms* or *phonetic signs*. In these signs the image carries no meaning whatsoever, being used by convention to represent one or more sounds of language. For example:  (*m*),¹  (*mn*) or  (*hpr*).
2. *Logograms* or *lexical signs*. They define the object of which they are an image. These signs may be accompanied by a vertical line that acts as a distinguishing mark, as shown in the case of , where  depicts a mouth and represents the word *r*, which means “mouth”.
3. *Determinatives* or *semantic signs*. These signs, which are not read, are placed at the end of a word to indicate that it corresponds to a given semantic group. They are of great importance since they allow us to differentiate between words that have the same consonantal representation but different meaning. For example, given the determinatives  (category *Writing - Abstract Notions*) and  (category *Human Being [Male]*), the sign  (*shj*) means “to write” in the case of  and “scribe” in the case of .


Moreover, the same glyph may belong to more than one of these categories at once. Symbols of the three categories are combined to form words and phrases.

2.2 Arrangement and Directionality

A noteworthy characteristic of Ancient Egyptian is the varied ways in which signs can be arranged to compose a text. Firstly, it is a continuous script, all words running together without dividers to separate words or phrases. This is not only characteristic of other ancient languages but also of contemporary ones such as Chinese or Japanese, where no word separators are used.

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 in *groups*, trying to fill the space available neatly, in a way which resembles current Hangul Korean script. This arrangement seems to be due to some kind of *horror vacui* or abhorrence of empty space.

Thus, the word “boat” was not written  (*dpt*), but

 instead.² To do this, they divided the space into imaginary ideal squares, one per sign group, with the size of the biggest symbol to be written. In turn, each of these imaginary squares could be divided into two or three horizontal or vertical *layers* or into four small *quarters*. Hieroglyphs were arranged within each group, distributing them into those divisions according to their shapes and intended sizes, trying to find the most harmonious and aesthetic arrangement.

Nor is the *direction of writing* unique. Hieroglyphic texts can be found written in horizontal rows, as with English and Arabic languages, for example, or in vertical columns, as

¹Where appropriate we will indicate, as in this case, the transliteration corresponding to the hieroglyphic text in question. The *transliteration* consists of representing the signs of a given writing system with those of another.

²As “described” in the title of this paper.



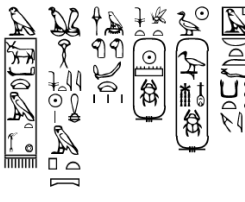

	left-to-right (⇒)	right-to-left (⇐)
H		
V		

Table 1: Writing directions: horizontal rows vs. vertical columns and left-to-right vs. right-to-left (examples taken from [15]).

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. In order to know how to read them, we must check in which direction the signs are looking. If they are facing left, we have to read from left to right; if they are facing right, we will read them from right to left. Table 1 shows several examples.

2.3 Cartouches



In Ancient Egyptian, a *cartouche* is a special symbol consisting of an oval ring around a text. It represents a loop of rope with a knot at one end and indicates that the text enclosed is a royal name. For instance, the following cartouche corresponds to Queen Cleopatra:



3. ENCODING HIEROGLYPHIC TEXTS

When working with hieroglyphic texts, contemporary scholars needed a practical way to represent them without having to re-draw their signs. The solution consisted of encoding these texts using regular characters. Firstly, in the case of handwriting and printing, the so-called *Gardiner’s List* provided Egyptologists with such a tool. Decades later, in the computer era, the need for digitalizing these texts and storing them as digital documents was solved with the appearance of the so-called *Manuel de Codage*. Next, both of them are introduced to the reader.

3.1 Gardiner’s List

One of the major contributions of Sir Alan Gardiner, one of the most important Egyptologists of the past century, was the so-called *Gardiner’s List* [5].³ In this list, considered a standard reference in the study of Ancient Egyptian hieroglyphs, the author classifies its signs into 26 categories according to their drawing, each one identified with a letter: category A corresponds to “*Man and his occupations*” (); B to “*Woman and her occupations*” (); etc. In turn, signs within each category are numbered sequentially. Thus, a given hieroglyph can be coded using the

³A complete on-line version is available at: http://en.wikipedia.org/wiki/Gardiner's_sign_list.






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

Table 2: Sign arrangement operators with MdC.

letter of its category and its number within the group. For example, the code G5 corresponds to sign  (“falcon”), the fifth element of category G (“Birds”). This classification includes the most common hieroglyphs (743 signs and 20 variants), enabling us to encode the majority of texts.

3.2 Manuel de Codage

Before the 80s, computers were rare, but by that time the use of computers started to spread to businesses, universities and even homes. Thus, in 1984 the first “*Table Ronde Informatique et Egyptologie*” of the *International Association of Egyptologists (IAE)*⁴ took place. During this meeting, it was decided to form a committee with the aim of designing and developing an encoding system for the digitalization of hieroglyphic texts. The resulting document, known as *Manuel de Codage (MdC)*, was published four years later [2] and constitutes the current standard encoding system used in Egyptology for the digitalization of hieroglyphic texts.

The MdC can be viewed as an evolution of the prior *Gardiner’s List*, where new codes and rules have been added for the accurate representation of hieroglyphs by using ASCII text. This way, individual signs are represented using their corresponding *Gardiner’s List* code or their corresponding phonetic value. In addition, new elements were included for encoding other features of the language [19, 2]. Next, we will introduce an overview of these additions.

3.2.1 Arrangement and Cartouches

Firstly, Table 2 presents the basic operations available in the MdC for managing the *arrangement of signs*. In order of precedence, these operators are: ‘-’ for concatenating adjacent signs; ‘:’ for stacking signs one over another; ‘*’ for placing two signs next to each other within a group; and *brackets* for grouping.

Moreover, the presence of a *cartouche* is indicated by enclosing the corresponding signs between ‘<’ (opening) and ‘>’ (closing) symbols, as in our previous example corresponding to Queen Cleopatra:

<-N29:E23-M17-V4-Q3-G1-D46:D21-G1-X1:H8->



3.2.2 Damaged Texts

One of the specific problems to be faced in this context was the representation of damaged texts in the most informative way. Over the millennia, the majority of hieroglyphic texts have disappeared and, if not, they have suffered the effects of time, exposure, vandalism, etc. Maybe our papyrus scroll is partly disintegrated, or part of the wall where the

text had been chiseled has collapsed or has been eroded beyond recognition, for example. This matter was solved by the use of the so-called *shades*, which allows us to express whether the sign or even its presence is recognisable or not, how many signs are affected, which parts of the symbols are damaged, etc. When the signs of the group are recognisable, the group space is divided into four numbered quarters, thus allowing us to indicate which of them are damaged by using the symbol ‘#’ and listing the quarters affected. If the damage spreads along several sign groups, it can be also represented by enclosing that text between ‘#b’ (opening) and ‘#b’ (closing) marks. When a sign group, or part of it, is completely lost, it can be denoted by using several ‘/’-type signs available: ‘//’ for denoting an entire missing group, ‘/’ for a missing quarter, ‘h/’ for a missing horizontal layer and ‘v/’ for a missing vertical half. Table 3 shows several examples of their use.

3.2.3 Text Segmentation

As explained in Section 2.2, Ancient Egyptian is a continuous script, with no dividers to separate words or phrases. However, when reading or writing these texts, scholars are able to recognize the words and sentences contained in them. This situation has been considered in the MdC code, which establishes that a single space ‘ ’ or underscore character ‘_’ can be used as *word delimiter*, whereas a double space ‘ ’ or underscore ‘__’ can be used as *sentence delimiter*. Unfortunately, since these separators have no effect on the graphical representation of the hieroglyphic text, most scholars do not use them when codifying their documents.

On the other hand, symbols ‘!’ and ‘!!’ are used as *end-of-line* and *end-of-page* markers to insert line breaks and page breaks within hieroglyphic texts. In this case, as they have a real effect on the output display, they are used.

3.2.4 Non-Hieroglyphic Text

Usually, when editing a hieroglyphic text, there is a need to combine hieroglyphs, transliterations, translations and other types of annotation within the same text. The MdC includes encoding support for this case. For this purpose, it assumes that all text is hieroglyphic unless it is enclosed between the symbols ‘+t’ (opening) and ‘+s’ (closing) in the case of transliterations, or ‘+l’ (opening) and ‘+s’ (closing) in the case of regular unformatted Latin text, for example.

4. SYSTEM REQUIREMENTS

Our purpose is to develop an IR system capable of operating on Egyptian hieroglyphic texts. Besides studying the basic nature of this language (see Section 2), our first step consisted of consulting an expert Egyptologist. This way, we could better understand the domain, how Egyptologists work with hieroglyphic texts and so extract the requirements of the system from the point of view of its potential future users and its application field.

Next, we introduce the basic features to be covered by the system:

1. **Simplicity:** The system should be intuitive and easy to use, with a minimum learning curve.
2. **Content indexing:** The system must be able to index documents containing conventional text and hieroglyphic text (either combined or separately) transparently to the user. At first we will focus on those

⁴<http://www.iae-egyptology.org/>

Description	Example
Q1 damaged but readable	X1*X1:Aa1*Aa1#1
full sign group damaged but readable	X1*X1:Aa1*Aa1#
sequence damaged but readable	A1-#b-A2-A3-#e-A4
Q1 sign missed but detectable	/*X1:Aa1*Aa1
upper half group missed	h/:Aa1*Aa1
sequence of two missed but detectable groups	A1-//-//-A4
sequence of missed undetectable groups	A1-#b- . . . -#e-A4

Table 3: Representing damaged text with MdC.

documents containing hieroglyphic texts written using the JSESH editor. Thus, we will be covering most of the digitalized contents currently available.

3. **Querying using MdC encoding:** In the case of hieroglyphic texts, the user will input the query using MdC alphanumeric encoding, with which he is familiar due to his everyday work.
4. **Display the query using glyphs:** Even if the user knows MdC well, this encoding consists of a confusing and complex sequence of alphanumeric codes where it is easy to make mistakes. Therefore, in order to make it easier to the user, the system should also display, in parallel, those pictograms corresponding to the input MdC query.
5. **Querying using Latin text:** Since the document collection will contain both hieroglyphic and conventional Latin text (e.g. English text), we also want to be able to submit conventional queries written in Latin text.
6. **Submission of mixed queries:** Given the mixed nature of the documents of the collection, the user will find useful it to make “mixed” queries combining both hieroglyphic and Latin text at the same time.
7. **Relevant documents retrieval:** Once the system obtains the set of documents that are relevant to the query, they will be presented to the user.
8. **Display of document contents:** The user should be able to access the content of those relevant documents retrieved by the system and check why they have been retrieved.

5. DESCRIPTION OF THE SYSTEM

According to the previous requirements, we have developed an IR system. As shown in Figure 1, which presents a schematic representation of the system and the processes involved, the general architecture and behavior of the tool corresponds to a classic Text Retrieval system. So, we can distinguish two different main phases: firstly, the indexing of the document collection on which you want to perform searches and, secondly, the querying and retrieval process.

5.1 Indexing

As shown in Figure 1, in this first phase the system accesses the documents of the input collection and extracts their contents. This content will be analyzed in order to separate the Latin text and the hieroglyphic text. Each kind of text will be normalized separately in order to generate their associated index terms for later indexing. Next, we describe the different processes involved in this phase and their associated modules.

5.1.1 Content Extraction

The indexing engine works on text documents, although their input format may be varied: .ODT, .DOC, .PDF, etc. The purpose of this first module of the system is to extract the text contained in those documents. It makes use of the Apache Tika toolkit,⁵ a software tool which can detect and extract both text and metadata from a wide range of different file types.

5.1.2 Text Preprocessing

Even within the same document, the conventional text content and the hieroglyphic text content need to be separated since they require different further processing. Thus, after extracting the content of a document, this content is preprocessed in order to separate both types of texts and to filter out useless data. For this task the system applies a *pattern matching* approach. For instance, in the case of detecting and extracting pieces of text corresponding to unformatted Latin text, the system uses a regular expression for identifying sequences of characters enclosed between the marks '+l' and '+s' since, as previously explained in Section 3.2.4, these are the tags defined in the MdC for delimiting regular unformatted text.

5.1.3 Text Normalization

This component focuses on processing the obtained text by applying a series of *text operations* for tokenizing, conflating and generating its associated index terms. 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 the well-known Apache Lucene search engine library.⁶

In the case of the Latin text processor, a standard normalization processing is performed [9]. Firstly, a standard lexical analysis is applied on the text for tokenizing it and

⁵<http://tika.apache.org>

⁶<http://lucene.apache.org/core/>

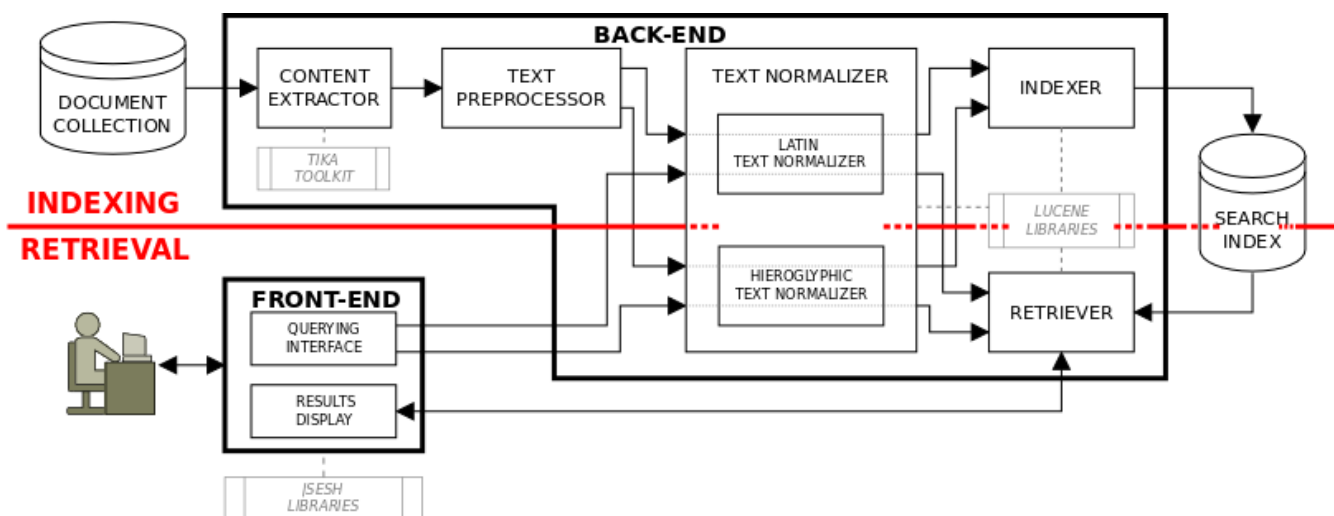



Figure 1: Schematic representation of the system: indexing and retrieval processes.

the resulting terms are then conflated by lowercasing them and removing both stopwords and diacritics.

However, the case of hieroglyphic text is completely different and presents several complications due to the peculiarities of this ancient writing system, as previously described in Section 2.

The first problem is that it is a continuous writing in which no space or symbol is used to separate words or phrases. As explained before in Section 3.2.3, although the MdC standard does provide the user with several symbols for that purpose, unfortunately they are not used in practice by most scholars in their transcriptions.

So, in this first distribution of our system we have opted to use sign *groups* as our working unit (i.e. those sign sequences delimited by '-' in MdC encoding). For example, the word  (D46:Q3*X1-P1) is composed by four signs but only two groups. Thus, the input text is tokenized in *groups*.

Additionally, unlike the Latin text processor, input text will not be lowercased this time, since MdC encoding is case-sensitive. This is also the case of punctuation marks, which form part of MdC encoding and provide information about the composition and arrangement of hieroglyphic texts (see Sections 3.2.1 and 3.2.2).

5.1.4 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 obtained during the normalization process will be indexed together with their occurrence positions within the text. For the implementation of this module we have also made use of Apache Lucene.

5.2 Querying and Retrieval

In this second phase we can distinguish, in turn, two main sub-processes, the querying process and the retrieval process, which are controlled by the user through his front-end interface.

5.2.1 Querying

As stated in the requirements, the system enables the user to query the indexed collection by using either hieroglyphs,

Latin text or a combination of both (*mixed queries*) —i.e. the user ask for documents containing, at the same time, both the Latin text terms and the hieroglyph sequence he has specified.

The query normalization process varies depending on the type of text and is parallel to that performed during the indexing process (Section 5.1.3). In the case of hieroglyphic text, two search modes are available at this time: *exact matching*, where we require the documents to be retrieved to contain exactly the same sign group sequence specified in the query (i.e. the same signs in the same arrangement for all the sign groups); and *approximate matching*, which allows the user to sub-specify the composition of a sign group (e.g. to require that a given group of the sequence contains the sign X1 but without specifying whether it contains any more signs or their arrangement within the group).

5.2.2 Retrieval

Once the query has been normalized, the recovery module accesses the index looking for matches and identifies those documents of the collection that are relevant to the query. The current implementation combines two retrieval models for scoring and ranking the documents [9]: firstly, the relevant documents are selected by using a Boolean model and, then, a Vector Space model is used for ranking those documents.⁷ The resulting list of documents is returned and presented to the user. Moreover, if required by the user, its content can be displayed and the matching terms shown highlighted to check why the document has been retrieved.

5.2.3 Front-End User Interface

One of the most interesting features of this part of the system is the design of the user interface. It seeks simplicity and clarity to make its use as easy and intuitive as possible, aiming at minimizing the learning curve of the user.

With regard to the input of the query, the system provides the user with separated search forms for Latin and hieroglyphic text, as can be seen in the right-hand panel of Figure 2. As stated in the requirements, in the case of

⁷See https://lucene.apache.org/core/3_5_0/api/core/org/apache/lucene/search/Similarity.html for further details.

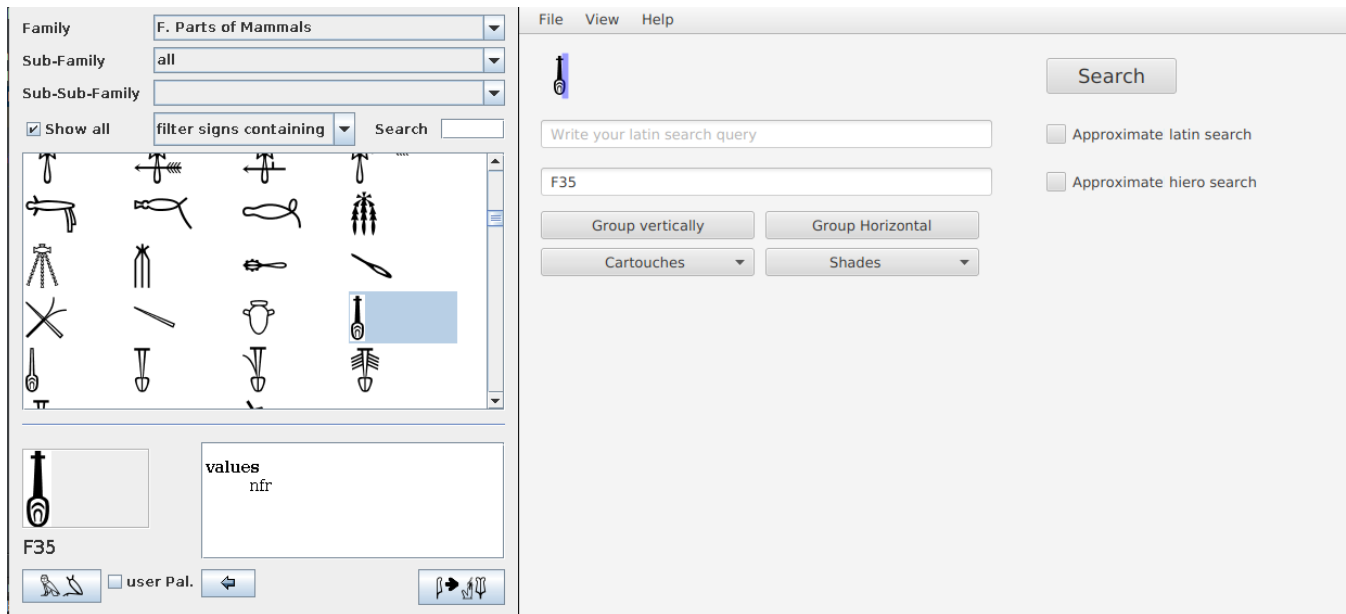


Figure 2: Querying the system using hieroglyphs.

hieroglyphic text queries —i.e. written using MdC—, the pictograms corresponding to the text being written will be displayed so that the user can check them. Moreover, in the case of hieroglyphic text queries and in addition to the original requirements, 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 symbols that allow the user to add them to the query by clicking on them, as shown in the left-hand panel of Figure 2. This palette also functions as a catalog of symbols and their variants, organized according to Gardiner’s List classification (see Section 3.1), so the user can navigate through it and consult the information and variants associated with each symbol. It also provides several options for handling the hieroglyphic text, such as adding shadows, creating cartouches of different styles or creating your own palettes.

For the implementation of this front-end we have made use of the libraries provided with the JSesh editing tool [16], including its symbol palette. As will be explained in Section 6, JSesh is, currently, the most popular editing tool among the Egyptology community. So, the user of our system will find an interface with a very similar appearance and behavior to that of the tool he is already familiar with. This greatly facilitates its use, even in the case of novice users.

This front-end is also responsible for presenting the user with the result of the search. This way, the list of documents relevant to the query is displayed using HTML. Also, from there the user can access the content of the document, which will be displayed highlighting, if required, the term matchings found. For this purpose, we have again made use of the libraries provided with JSesh.

Finally, our interface supports *internationalization*. At this time the user can choose between English, French, Spanish and Galician.

6. RELATED WORK

From the beginning, hieroglyphic text processing in Egyptology has mostly focused on the development of classic-style text edition tools [6]. This was not a choice but a primary need. The reason is due to the fact that since there were no typewriters for hieroglyphs, Egyptologists had to rely on handwritten texts when writing and sharing documents, a practical limitation that could easily lead to misinterpretations. Moreover, in the case of books, the hieroglyphic texts printed in their pages were typographical transcriptions or even mere copies of those handwritten by their authors. Apart from the aforementioned possible misinterpretations, the extensive and complex typographical fonts required were very rare and made typesetting very hard. Thus, the need for hieroglyphic text processing software was peremptory [7].

Even before the appearance of the *Manuel de Codage* (MdC) standard encoding system in 1988 (see Section 3.2), one of its parents, Jan Burman, had created the text processing software GLYPH [6]. This initial tool, implemented in Fortran 77, was published for DOS operating system in 1986 and laid the foundations of future hieroglyphic text processors.

GLYPH subsequently evolved and migrated to other operating systems: MACSCRIBE for Macintosh and WINGLYPH for Windows (3.1 and 95). Both tools have been the most widely used editing software in the field of Egyptology until the beginning of this century.

A new tool, VISUALGLYPH,⁸ developed by Gunther Lapp, appeared in 2003. It integrated novel improvements that enabled users to freely position, size and rotate hieroglyphs. This allowed a more accurate representation of certain texts.

Currently, the most widely used word processor in Egyptology is, in all probability, JSesh, developed by Serge Rosmorduc [16]. Among its features, this tool offered backward compatibility, thanks to its ability to read files created with WINGLYPH. Moreover, it was the first free software of its

⁸<https://aegyptologie.unibas.ch/werkzeuge/visualglyph-for-pc/>

type, and its Java code is also freely available to users and developers. It is noteworthy to say that Mr. Rosmorduc had also previously developed HieroTeX, a L^AT_EX package for writing hieroglyphs [15]. In fact, it has been used to prepare this document.

At this point we should notice that although the MdC had been supposedly taken by these tools as a standard for the encoding of hieroglyphic texts, this was not completely true. All these editors took the MdC as their base but, at the same time, they established their own particular specifications. This meant that, with exceptions — as in the case of the backward compatibility of JSESH with WINGLYPH—, a text written with a given program could not be opened and edited with another one unless it has been previously rewritten in the new notation. This fact makes it difficult to share documents between researchers and establish common corpora [6].

7. CONCLUSIONS AND FUTURE WORK

Throughout this work we have presented an Information Retrieval system which can operate on Egyptian hieroglyphic texts, taking into account their peculiarities both at lexical and at encoding level. Our system admits queries containing both hieroglyphic and Latin text queries. Special care has been paid to the front-end interface in order to make it as intuitive and easy to use as possible.

To the best of our knowledge this is the first tool of its kind. We intend to release it under a free license for its use, for example, in Digital Heritage applications.

With respect to future work, the current initial system can be improved in several ways. Firstly, by extending its range of supported encodings and formats in order to accept as input source new types of documents. For example, documents created with WINGLYPH, Unicode text documents [18] or, as in the case of this article, L^AT_EX documents built using the package HieroTeX [15].

At this first stage our system has been configured to use a Boolean model for relevant document selection, which is based on exact group-sequence matching with the possibility of sub-specifying the sign group composition. Those selected documents are later scored and 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. Moreover, given the nature of hieroglyphic script and the noisy character of the input texts (often containing transcriptions of deteriorated texts), it would also be interesting to study the application of fuzzy matching algorithms.

After analyzing the case of several contemporary languages which share characteristics with Ancient Egyptian writing, such as Chinese, Japanese, Korean or Arabic, we have produced a preliminary implementation of a solution based on the use of sign group n -grams as a working unit [4, 12, 8, 11]. An n -gram is a consecutive sub-sequence of n elements, groups of signs in this case, from a given sequence. For example, the term $\overline{\text{U7:D21-G43:X1-A2}}$ is composed of five signs arranged into three groups. So, if we split it into group bigrams (i.e. n -grams with $n=2$) we would obtain as output: $\overline{\text{U7:D21-G43:X1}}$ and $\overline{\text{G43:X1-A2}}$. Moreover, the use of n -grams as a working unit in IR tasks has multiple advantages in terms of simplicity, robustness, and language and domain independence [20].

The application of phonetic matching [13] or even conflation mechanisms based on lemmatization or morphological analysis [14] are also alternatives to be considered. However, it would also require a further analysis and discussion with the users about the evaluation process to be applied to assess the results.

Finally, from a more practical point of view, another possibility to study is to migrate the system from the current desktop application to a client-server (possibly even web-based) application.

8. ACKNOWLEDGMENTS

This research has been partially funded by the Spanish Ministry of Economy and Competitiveness (MINECO) through 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 the Ancient Egyptian language and acting as our fictional client. We would also like to thank Dr. Serge Rosmorduc, Associate of the Conservatoire National des Arts et Métiers (CNAM) for all his support when working with JSESH.

9. REFERENCES

- [1] J. P. Allen. *Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs*. Cambridge University Press, Cambridge, UK, 2000.
- [2] 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*, volume 8 of *Mémoires de l'Académie des Inscriptions et Belles-Lettres*. De Boccard, Paris, 1988.
- [3] J. Cervelló-Autuori. *Escrituras, Lengua y Cultura en el Antiguo Egipto*. El espejo y la lámpara. Edicions UAB, Bellaterra (Cerdanyola del Vallès), 2015.
- [4] S. Foo and H. Li. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management*, 40(1):161–190, 2004.
- [5] A. H. Gardiner. *Egyptian grammar: being an introduction to the study of hieroglyphs*. Griffith Institute, Ashmolean Museum, Oxford, 3rd ed., 1957.
- [6] R. Gozzoli. *Texts, Languages & Information Technology in Egyptology: Selected Papers from the Meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie), Liège, 6-8 July 2010*, volume 9 of *Collection Ægyptiaca Leodiensia*, chapter Hieroglyphic Text Processors, Manuel de Codage, Unicode and Lexicography, pages 89–101. Presses Universitaires de Liège, Liège, 2013.
- [7] N. Grimal. Hiéroglyphes et ordinateurs. *BRISES. Bulletin de Recherches sur l'Information en Sciences Économiques Humaines et Sociales*, (15):57–60, 1990.
- [8] J. H. Lee and J. S. Ahn. Using n -grams for Korean text retrieval. In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 216–224, New York, NY, USA, 1996. ACM.

- [9] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
- [10] W. McCarty. *Encyclopedia of Library and Information Science*, chapter Humanities Computing, pages 1224–1235. Marcel Dekker, New York, 2nd edition, 2003.
- [11] 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)*, 55(11):1002–1007, 2004.
- [12] Y. Ogawa and T. Matsuda. Overlapping statistical segmentation for effective indexing of Japanese text. *Information Processing and Management*, 35(4):463–480, 1999.
- [13] D. Pinto, D. Vilariño, Y. Alemán, H. Gómez, and N. Loya. The Soundex Phonetic Algorithm Revisited for SMS-based Information Retrieval. In *II Congreso Español de Recuperación de la Información (CERI 2012)*, number 11 in Col.lecció e-Traballs d'Informàtica i Tecnologia, pages 97–108, Castelló de la Plana, Spain, 2012. Publicacions de la Universitat Jaume I.
- [14] M. Piotrowski. *Natural Language Processing for Historical Texts*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.
- [15] S. Rosmorduc. HieroTeX: A LaTeXperiment of hieroglyphic typesetting, 2003. Package available online at: <http://www.ctan.org/tex-archive/language/hieroglyph> (visited on April 2016).
- [16] S. Rosmorduc. JSesh documentation, 2014. Available online at: <http://jsesh.qenherkhopeshef.org/> (visited on April 2016).
- [17] UNESCO. Charter on the Preservation of Digital Heritage, 2003. Available in http://portal.unesco.org/en/ev.php-URL_ID=17721&URL_DO=DO_TOPIC&URL_SECTION=201.html (visited on April 2016).
- [18] Unicode 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 April 2016).
- [19] H. van den Berg. Manuel de Codage: A standard system for the computer-encoding of Egyptian transliteration and hieroglyphic texts. Centre for Computer-Aided Egyptological Research (CCER), 1997. Currently available at: <http://www.catchpenny.org/codage/> (visited on April 2016).
- [20] J. Vilares, M. A. Alonso, Y. Doval, and M. Vilares. Studying the effect and treatment of misspelled queries in Cross-Language Information Retrieval. *Information Processing and Management*, 2016. DOI: 10.1016/j.ipm.2015.12.010. (*In press*).