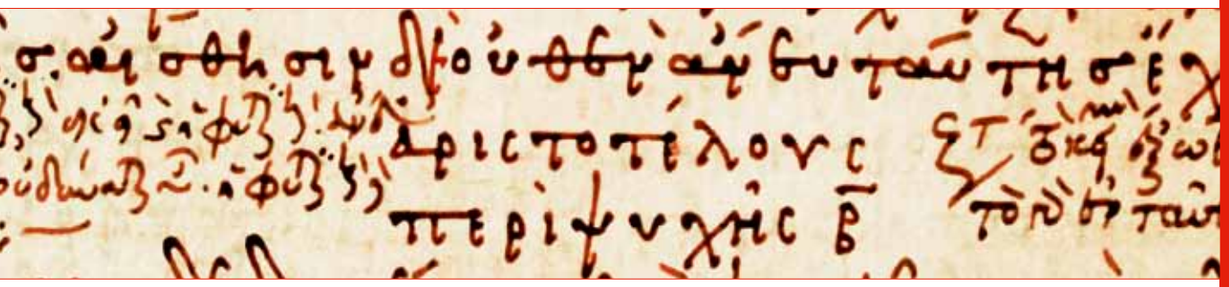
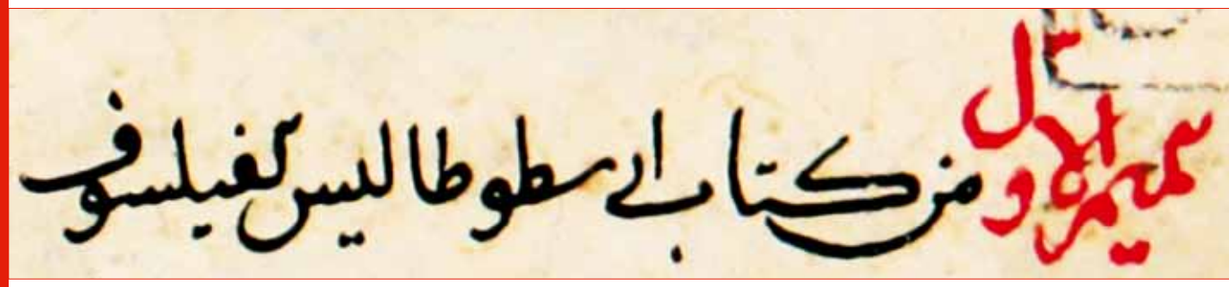
ISSN 2239-012X



With the support of the European Research Council

Studia graeco-arabica

Studia graeco-arabica



3

2013

Studia graeco-arabica

The Journal of the Project
Greek into Arabic
Philosophical Concepts and Linguistic Bridges
European Research Council Advanced Grant 249431

3

2013

Published by
ERC Greek into Arabic
Philosophical Concepts and Linguistic Bridges
European Research Council Advanced Grant 249431

Web site: http://www.greekintoarabic.eu
Service Provider: Università di Pisa, Area Serra - Servizi di Rete di Ateneo

ISSN 2239-012X

Registered at the law court of Pisa, 18/12, November 23, 2012.
Editor in chief Cristina D'Ancona.

Cover
Mašhad, Kitābḫāna-i Āsitān-i Quds-i Raḍawī 300, f. 1v
Paris, Bibliothèque Nationale de France, grec 1853, f. 186v

The Publisher remains at the disposal of the rightholders, and is ready to make up for unintentional omissions.

Publisher and Graphic Design
Pacini Editore
Via A. Gherardesca
56121 Ospedaletto (Pisa) - Italy

Printing
Industrie Grafiche Pacini

Studia graeco-arabica

3

2013

G2A Web Application

Istituto di Linguistica Computazionale “Antonio Zampolli”
Consiglio Nazionale delle Ricerche - Area della Ricerca di Pisa

Indexing techniques and variant readings management

Angelo Mario Del Grosso

Abstract
This paper illustrates indexing routines developed for the *G2A Web Application*, a philological system totally open source designed by the Team of the ILC-CNR of Pisa within the context of the ERC project *Greek into Arabic. Philosophical Concepts and Linguistic Bridges* (*Ideas* AdG 249431). Section 1 introduces the concept of 'index' in this peculiar field. The indexing process implemented by the ILC-CNR Team for the *G2A Web Application* is illustrated in Section 2. Section 3 discusses the component of textual criticism.

Introduction

'Index' is the most representative term of the intersection between the needs of philologists and computer scientists. In this paper, 'index' means an auxiliary structure able to ensure efficient access to information after an external request; the paper aims at exposing the transformation of 'index' from manual to automatic and from static to dynamic (indexing).[1] We focused on creating useful tools in order to organize and retrieve relevant data.[2] These tools must provide accurate results in shorter time, according to the methods and techniques developed by the specialists of information retrieval.[3]

Research, studies and formal approaches on the information, linguistic analysis and textual criticism literature date from the middle of the past century. The *Index Thomisticus*,[4] built from 1949 and completed thirty years later under the direction of Roberto Busa S.J. is considered as the first example of systemic work in computational linguistics. It allows the identification and storage of contexts where the words occur, and groups them in a list alphabetically ordered by form, lemmas, and frequency (indexes and concordances). More recent methods and applications are: lexicography and electronic thesauri; indexing generated by natural language processing (NLP);[5] automatic

---

[1] A static index is a structure difficult to modify and synchronize, while a dynamic index is able to rearrange its own data automatically.

[2] Data are a basic information structure disconnected from their own semantic context. They may be of various nature and assume various meanings. In this contribution, data and information are predominantly of textual type, therefore, when not otherwise specified, we refer to this category.

[3] An academic debate highlights that 'data', 'document', and 'text retrieval' should be different from 'information retrieval', see C.N. Mooers, *Making information retrieval pay*, Zator, Boston 1951. In this work such distinctions have not been taken into account. Racommended readings about a general overview concerning information retrieval are R. Baeza-Yates - B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology behind Search*, Pearson Education Limited, Harlow 2011[2]; C.D. Manning - P. Raghavan - H. Schütze, *Introduction to Information Retrieval*, Cambridge U. P., New York 2008; and G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, New York 1968.

[4] *Index Thomisticus* at URL: <http://www.corpusthomisticum.org> (accessed on March, 2013).

[5] Natural language processing (NLP) involves many disciplines aimed at investigating the cognitive and linguistic human processes in order to simulate them. An introduction to this fascinating subject could be carried out by reading, among others, D. Jurafsky - J.H. Martin, *Speech and Language Processing*, Prentice-Hall, Upper Saddle River 2009[2] and

extraction of meaning from a corpus (semantics); stylistic analysis of a text, or of an author. From the perspective of computer science and engineering, the index benefit is an outstanding practice to ensure a correct and consistent data organization, as well as efficient access to information.

Indexes improve data management giving the chance to human or software agents to record, store, and read sequences of information (i.e. bits or byte). Operating systems handle information thanks to homogeneous structures (files). They use indexes to manipulate files through the file system[6] (both logical in front-end[7] and physical in back-end).[8] It should be stressed that an index has two distinct purposes as the following list points out:

1) *back-end index*: it is exploited by the machine and indicates the "data structure"[9] as the internal representation of the stored information (it means how raw data are organized inside the computer persistence unit);

2) *front-end index*: it is exploited by the end-user and it indicates a uniform and ordered list of key-terms (which may be the chapter headings, paragraphs, word-forms of text or the lemmas extracted by means of linguistic analysis, etc.).

To date, the most promising methodologies and applications arise from the joint effort of computer science and linguistics, which in its turn may be seen as the attempt to overcome the problem of the unstructured and non-indexed information plethora created by the growth of the World Wide Web in the last ten years, and by hypermedia and multimedia publications. It is important to realize that the joint effort is also related to the uninterrupted growth of telematics infrastructure (Internet backbone).[10]

In this context, it goes without saying that data classification and information retrieval are strategic challenges both from the academic and economic point of view. Some important technological projects within the area of digital humanities have already reached momentous results: to name just a few, Interedition, Bamboo, Archive.org, Europeana, TextGrid, Clarin, Dariah give good examples of this development. They set their bases on the web paradigm and on a range of distributed "core of functionalities".[11] Automatic text analysis and processes for creating indexes are widely used to store and manipulate data for the purpose of searching and for scholarly activities.

Current trends amplify the collection and management of text in a much broader and accurate manner (storage issue). In the long run, the web will be multilingual, semantically processed, indexed

---

C.D. Manning -H. Schütze, *Foundations of Statistical Natural Language Processing*, Massachusetts Institute of Technology Press, Cambridge 1999.

[6] In this paper is not appropriate to deepen this topic; let me just recall that one of the main components of a computer is the operating system. It deals with the control and management of all hardware devices and with the interaction between the application software and the user. This component holds a structure (that can be seen as an index) for the logical organization of files, called file system.

[7] The term 'front-end' indicates the structures by which the end user interacts.

[8] The term 'back-end' is referred to the programs and data structures which are not directly accessible by the user.

[9] The term 'data structure' means how information is organized internally to a computer, more formally "a data structure is a set of domains D, a set of functions F, and set of axioms A. This triple (D, F, A) denotes the data structure d" see A.A. Puntambekar, *Advanced Data Structures & Algorithms*, Technical Publications, Pune 2008, in part. p. 39.

[10] The backbone of Internet is the interconnection of the main Net devices. It is now made up of millions of computers able to communicate at high speed (rate). The order is the Gigabit per second (Gbit/s), speed reachable through the fiber-optic wiring infrastructure. The storage capacity is quantifiable in millions of gigabytes distributed in huge data centres (it is not easy to estimate how much information is stored in the 'cloud', but the order is trillions of gigabytes [source Cisco]).

[11] The word 'distributed' indicates, in computer field, a set of objects interconnected each other which cooperate in order to achieve common goals.

and interconnected. It will be reachable in a large-scale by user-friendly and flexible systems. Nowadays data are collaboratively produced and processed in real time through heterogeneous devices (cross-devices) and different software applications.

The G2A platform fits into this frame and sets for itself the task of obtaining a high degree of flexibility, versatility and modularity of the components. The target set is obtained starting from data retrieval and the practices of information extraction. The G2A functionalities are the "core" of a web platform oriented to the analysis of parallel texts. Designed and developed at the Institute of Computational Linguistics (ILC-CNR), this application provides software modules for texts translated from an ancient language into another. In our case study, namely the Arabic version of Plotinus' *Enneads*, the Arabic language is the adapted translation of a text originally written in Greek. The index is built by using:

- an efficient and effective open source software[12] library (LUCENE)[13] for indexing textual documents;
- a morphological engine devoted to aid users for their linguistic analyses;
- methods and algorithms for string alignments;
- a native XML[14] system for data persistence (eXist-db).[15]

Digital and computational methodologies and approaches designed to handle philological and literary issues for scholarly textual edition have to deal with a broad range of related areas, such as multimedia capture and representation, compression, information coding of data and metadata. Indexing, linguistic analyses, and philological annotations are the basic tools which can find a practical and useful application.

The projects and research groups addressing this issue and directing their efforts towards the study and support for textual criticism[16] are quite few, although the label "computational philology" is not of recent use[17] and its first formalization goes back to an article published at the beginning of the nineties of the past century.[18] In order to interoperate and share data, G2A modules and components have to deal with information units coded using standard representations. The formalism and the guidelines of the Text Encoding Initiative (TEI),[19] expressed by the markup language XML, is the starting point to the expected goals.

---

[12] A software library is a structured collection of features which are invoked by appropriate method calls exposed to developers (API, Application Programming Interface).

[13] Lucene Web site at URL: <http://lucene.apache.org> (accessed on March, 2013).

[14] XML stands for eXtensible Markup Language. It is a W3C standard language realized originally from SGML language, for a simple and standard way to get structured documents. See also the XML w3c Web site at URL: <http://www.w3.org/XML/> (accessed on March 2013)

[15] eXist-db Web site at URL <http://exist-db.org> (accessed on March 2013).

[16] F. Gibbs - T. Owens (eds.), "Building Better Digital Humanities Tools: Toward broader audiences and user-centered designs", *Digtal Humanities Quarterly* 6/2 (2012) [URL: <http://www.digitalhumanities.org/dhq/vol/6/2/000136/000136.html>]; A. Badeu, "Rome Wasn't Digitized in a Day: Building a Cyberinfrastructure for Digital Classicists", *Council on Library and Information Resources Publication* (2011) [URL: <http://www.clir.org/pubs/reports/pub150/reports/pub150/pub150.pd>].

[17] The expression "computational philology" has been used for the first time in 1968 inside a computer science document about computational linguistic, see S. Kuno - A.G. Oettinger, "Computational Linguistics in a Ph.D. Computer Science Program", *Comm. ACM* 11, 12 (1968), in part. p. 835.

[18] A. Bozzi, "Towards a Philological Workstation", *Revue informatique et statistique dans les Sciences humaines* 29 (1993), p. 33-49.

[19] TEI Web site at URL: <http://www.tei-c.org/index.xml> (accessed on March 2013).

G2A parallel indexing

One of the objectives of the G2A platform is to allow users to identify and locate relevant parallel contexts (also called contrastive concordances).[20] The indexing process is the backbone of the whole architecture; once applied to the available[21] documents, it provides successive steps, as follows:

(1) parallel pericopes segmentation;
(2) text analysis;
(3) construction of indexes;
(4) storage and information management.

1. Parallel pericopes segmentation

The text resources, both in Greek and Arabic, stored in electronic[22] format (Tab. 1), have been segmented into uniform fragments (pericopes)[23] in order to create parallel textual units ruled by consistency (meaning for consistency a segmentation and a successive connection among parallel textual segments based on multiple aspects, such as semantic or/and linguistic observations, about which scholars operate thanks to the annotations recorded in the system). The granularity of the division is such to obtain benefits for the automatic analysis of resources as well as for individual scholarly analysis.

The pericopes (Tab. 2) have an identification number (univocal) both for the Arabic and the Greek text; a third number connects these identifiers (1). It determines the relation between two textual chunks.

$$Id_{pair} = f(<id_{ar}, id_{gr}>) \qquad (1)$$

The Arabic text is a free translation that at times verges on adaptation of the original Greek text; it often attests misalignments and transpositions. This leads to the definition of four different types of pericopes:

1. Greek-Arabic pericopes pair: the Arabic textual segment has the relevant text chunk in the Greek pericope (i.e., textual segments follow the original text flow for both languages);
2. Greek degenerate[24] pericopes pair: the Arabic pericope has been linked with a Greek one which has no text;
3. Arabic degenerate pericopes pair: conversely, the Greek pericope has an Arabic parallel segment without text;
4. Transposed pericopes pair: parallel textual segments, in this particular case, are not aligned (i.e., the Arabic pericopes do not follow the original text flow in the Greek work).

---

[20] A. Bozzi, *Il trattato ippocratico Riguardo all'aria, all'acqua, allo spazio e la sua traduzione latina tardo-antica. Concordanze contrastive con il calcolatore elettronico e commento linguistico-filologico al lessico tecnico latino*, Giardini, Pisa 1981.

[21] The linguistic and textual analyses have been carried on the fourth, fifth and part of the sixth book of the *Enneads* of Plotinus (for Greek) and on the pseudo-*Theology* of Aristotle (for Arabic).

[22] The electronic format of the digitized texts follows the standard Unicode (for character encoding and the internal representation of the file i.e. low level), and the standard TEI for the XML textual data encoding (structured representation i.e. high level).

[23] About this topic see: Bozzi, *Il trattato ippocratico Riguardo all'aria, all'acqua, allo spazio e la sua traduzione latina tardo-antica*; F. Boschetti, "Strumenti per l'analisi di testi bilingui al servizio dell'epigrafia digitale", *Lexis* 31 (2013), in print.

[24] The word 'degenerate' has been taken from the mathematical field and indicates "a limiting case in which a class of object changes its nature so as to belong to another, usually simpler, class". A classical example is a circle (geometric shape) that collapses into a point, see G. Arfken, *Mathematical Methods for Physicists*, Academic Press, Orlando 1985^3, p. 513-14.

Tab. 1. Example of TEI-XML texts encoding.

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI.2>
        <teiHeader>
                <fileDesc>
                        <titleStmt>
                                [...]
                        </titleStmt>
                </fileDesc>
        </teiHeader>
        <text>
                <body>
                        <div1 resp="ed">
                                <head>Badawi ONE</head>
                                <pb n="3" scan=" " />
                                <p>
                                        <hi rend="italic">بسم الله الرحمان الرحيم </hi>
                                        <lb />
                                        <hi rend="italic">الحمد لله رب العالمين، والصلاة على محمد
                                        وآله</hi>
                                </p>
                                <p>
                                        <hi rend="italic">الميمر الأوّل </hi>
                                        <lb />
                                        <hi rend="italic">من كتاب أرسطاطاليس الفيلسوف</hi>
                                        <lb />
                                        <hi rend="italic">المسمّى باليونانيّة "أثولوجيا"</hi>
                                </p>
                                <p>
                                        <hi rend="italic">وهو قول على الربوبيّة، تفسير فرفريوس الصوري</
hi>
                                        <lb />
                                        <hi rend="italic">ونقله إلى العربيّة عبد المسيح بن عبد الله بن ناعمة
                                        الحمصي</hi>
                                        <lb />
                                        <hi rend="italic">وأصلحه، لأحمد بن المعتصم بالله، أبو يوسف
                                        يعقوب</hi>
                                        <lb />
                                        <hi rend="italic">ابن إسحاق الكندي رحمه الله</hi>
                                        <lb />
                                </p>
                                <p>
                                        جدير بكلّ ساعٍ لمعرفة الغاية التي هو عامدها – للحاجة اللازمة
                                        إليها
                                        <lb />
                                        وقدر المنفعة الواصلة إليه في لزومه مسلك البغية  تدميث لها –
                                        تدميث الأساليب القاصدة
                                        <lb />
                                        إلى عين اليقين المزيل للشكّ عن النفوس عند الإفضاء به إلى ما طلب
                                        منها، وأن
                                        <lb />
                                </p>
                        </div1>
                </body>
        </text>
</TEI.2>
```

Snippet of XML encoding using a subset of the elements supported by the Text Encoding Initiative. The source digitized refers to ʻA. Badawī, *Aflūṭīn ʻinda-l-ʻArab*, Dār al-Nahḍat al-ʻarabiyya, Cairo 1966, p. 3.

Tab. 2. Greek-Arabic pericopes pair encoding.

```
<?xml version="1.0" encoding="UTF-8"?>
<add>
        <doc>
                <field name="id">542</field>
                <field name="pericope_ar">ولا يبقى في موضعه الأوّل، لأنّه يشتاق إلى الفعل كثيرًا وإلى زَيْن
                 الأشياء التي رآها في العقل، </field>
                <field name="pericope_gr">καὶ κοσμεῖν ὀρεγόμενον καθὰ ἐν νῷ εἶδεν,</field>
                <field name="id_ar">19.0304</field>
                <field name="id_gr">542</field>
                <field name="info_ar">I, p. 19.3-4</field>
                <field name="info_gr">IV 7[2], 13.6</field>
                <field name="note" />
        </doc>
</add>
```

This table shows an example of parallel pericopes encoded in XML. The eXtensible Markup Language (XML) is composed by elements (called also tags) which describe and characterize data. Tags are formed by a couple of angular brackets which define the name of the element (< *element-name* >). Tags can have attributes which assume specific values included in quoted strings (< element-name attribute-name="*attribute-value*">). Tags mark data as they are placed before (open tag) and after (close tag) the phenomena to be annotated (the close tags differ from the open ones as they have a backslash just before the element name and they have no attributes, i.e. </*element-name*>). Elements can be nested one inside the other, forming in this way a hierarchical structure ('tree structure'). The example above consists of the following elements: (I) add (meaning the addiction of a pericopes pair in the system), (II) doc (meaning the document, in this case the pericopes pair), and (III) fields (meaning the data and metadata linked to the pericopes pair). Field tags have attributes in order to code different kinds of information: (a) id; (b) pericope_ar; (c) pericope_gr; (d) id_ar; (s) id_gr; (f) info_ar; (g) info_gr; and (h) notes.

The ID field (identifier/identification number) denotes a univocal identification number of the pair. The formula (1) is written as an identity function[25] (2), which means that the identification number of the pair assumes the value of the Greek text identifier (Id_gr) and takes increasing and unique[26] values for each pericope.

$$Id_{pair} = id_{gr} \qquad (2)$$

The identification number of the Arabic pericopes (id_ar) follows a different structure. The number consists of three parts: (i) the page number of the original[27] text, (ii) a first couples of decimal values indicating the starting rows of the pericope in the page, and (iii) a second couples of decimal values indicating the ending rows (e.g., 55.1417 indicates that the pericope is selected on page 55 of the edition, from line 14 to line 17). Special attention has been paid when the Arabic pericope "overtakes" a page. In this special case, the second couple of decimals assumes a value that is smaller if compared to the first couple (e.g., 155.2201 indicates that the pericopes begins at line 22 on page 155 and ends at the first line of page 156).

[25] The identity function is the function which assigns every real number to the same real number. Cf. M. Petkovšek - H.S. Wilf - D. Zeilberger, "Identities", in A.K. Peters (ed.), *A=B*, Wellesley MA, USA 1996, p. 21-3.

[26] The pairs, starting from the value 1 (one), have an identification number (id) equal to a natural progressive number (such as, 1, 2, 3, 4, ...).

[27] Badawī (ed.), *Aflūṭīn ʿinda-l-ʿArab* (see above).

Pericopes without text (i.e., the degenerate pericopes) have a special treatment:

1) In case of *Greek degenerate pericopes pair*, the identification number (ID) brings also a decimal part (e.g., the pericope pair with id = 1.01 indicates that there is only the Arabic text and is located between the two Arabic-Greek pericope pairs with id = 1 and id = 2). In this way we can put several pericopes between two consecutive Greek-Arabic parallel segments.

2) In case of *Arabic degenerate pericopes pair*, the value of the identification number (ID) follows the natural progressive course (cf. note 26). The Arabic pericope identifier (Id_ar) takes values greater than the last page number. The whole page number of the reference book is 240, therefore the first Arabic degenerate pericope has an ID 241, the second one has an ID 242, and so on).

The system allows the alignment of parallel pericopes based on the original text flow, both Arabic and Greek. Through the assignment of identifiers to each textual segment as defined above, if the reading is led by Greek, a leap over the line, in the corresponding Arabic text, shows that the Greek pericope was transposed compared to the Arabic text tradition (Fig. 1): in the Arabic version, the Greek pericope has been transposed. Users are able to get visually what portion was transposed, thanks to the possibility offered by the system to rearrange the text according to the Arabic flow. Text division into parallel and homogeneous segments was also performed to facilitate indexing and text analysis. Indeed, elements have a well-defined indexing unit (i.e., the pericope), and the contexts are available in the pericopes pair.

Fig. 1. Parallel pericopes with transposed text.

The example shows a text transposition in the Arabic text. In the last colum of the figure the identification numeber of the Arabic pericope is 55.0305 and means that the corresponding textual segment belongs to the page number 55 of the reference edition. The pericope starts at line three and ends at line five. The subsequent pericope has 55.1011 as the identification number, which means that the corresponding text starts at line ten and ends at line eleven. It is clear that from line six to line nine a gap occurs. The reason of this gap is the transposition of a part of the text.

2. Text analysis

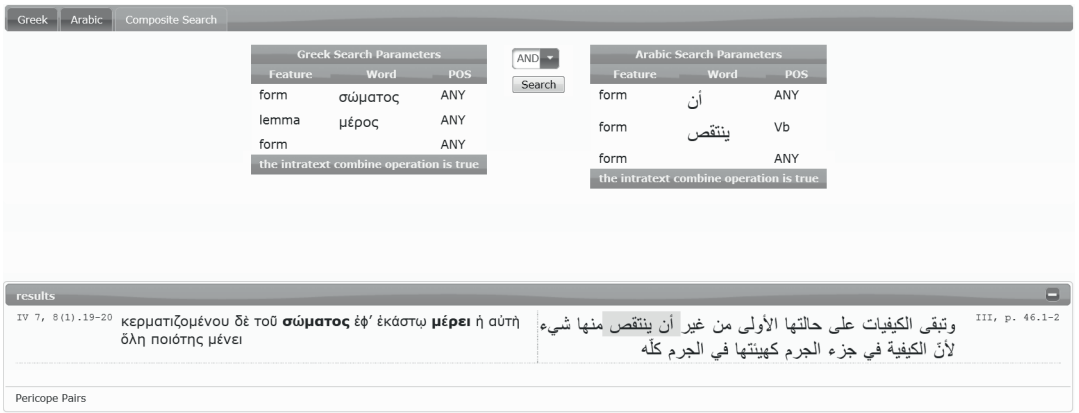
The analysis process has two main objectives devoted to increase the digital resource significance[28] and availability in order to:

(1) build a systematic index which allows to quickly perform search operations;

(2) associate to each linguistic unit a morpho-syntactic, and semantic information and other remarks.

We can send specific requests to the system (query)[29] by improving the information related to the text, and get the relevant results in a reasonable time.[30] We trace the following pseudo-syntax as an example of advanced query that users might submit: *term_Gj NEAR_3 lem_Gk AND term_Aj NEAR_1 verb_Ak*. This procedure (Fig. 2) means that users can search for all pairs of pericopes which have a given term attested in Greek (*term_Gj* might assume, for example, the word "σώματος"), next to a given lemma (*lem_Gk* might assume the word "μέρος") distant three words (*NEAR_3*) from the first term (σώματος), and having (the pericopes pairs which I am looking for), inside the Arabic text segment (the Arabic pericope), a precise term (*term_Aj*, for example "أن") followed by a particular verb (*verb_Ak*, for example "ينتقص").[31]

Fig. 2. Advanced search performed through G2A platform.



The figure shows a combined search taking into account both languages (Greek and Arabic) and linguistic features (such as lemma or verb filtering). Users are able to search for parallel periscopes having in Greek text the word form σώματος near the lemma μέρος and having in the Arabic counterpart the word form أن near the verb ينتقص.

The process implements several specific activities:

a) tokenization;

b) filtering/normalization;

c) linguistic annotation and processing.

---

[28] The term 'significance' expresses the actions taken on the document in order to eliminate the whole 'information noise', meaning neutralization of information irrelevant to index and classify documents.

[29] The term 'query' indicates the operation by which users ask the engine what they would get from the interrogation.

[30] The adjective (reasonable) is purposely qualitative and not quantitative, since the optimization of retrieval systems is in itself a line of research.

[31] The example is intended at demonstrating any correspondence between the Greek and Arabic words. The purpose is only to show the 'expressiveness' of the search functionality (the scholar is able to perform himself queries which are relevant for his domain).

The routines above can be used individually or to produce joint results. It is possible, in other words, to see these tasks as levels where data input derives from the under-level job and data output is formatted and structured so that it can be used by the top-level processes.

A token is a sequence of alphanumeric codes detected from the digital text. It is an independent and uniform elaboration unit.[32] Tokenization, therefore, is an activity devoted to identify tokens from a text stream. In most Western languages a token corresponds to a graphic form preceded and followed by a blank space or a punctuation character (e.g., a tokenization process performed on the chunk of text "Εἰ δέ ἐστιν ἀθάνατος", produces four tokens: respectively, the first has the value of [Εἰ], the second of [δέ], the third of [ἐστιν], and the fourth token is [ἀθάνατος]). Punctuation marks are treated as autonomous tokens (e.g., "ἕκαστος ἡμῶν, ἢ φθείρεται" produces five tokens: respectively, [ἕκαστος] [ἡμῶν] [,] [ἢ] [φθείρεται], where characters in square brackets represent the extraction process). The definition of "token" mentioned above is produced through heuristic approaches rather than through real scientific assessments. Thus, is not always universally accepted. It is, indeed, a source of problem for many linguistic and philological phenomena such as the correct handling of punctuation (e.g., in presence of abbreviations and acronyms), as well as when polyrematics and compound terms must be taken into account, or when words contain clitics elements, or also when one is faced with orthographic errors. In languages such as Suhaili or German and Arabic, which present agglutinating features, a whole sentence can be expressed by a single word. In these cases a tokenization algorithm has to include a segmentation phase in order to recognize the elaboration units kept in compound words (Tab. 3).

Tab. 3. Example of a token segmentation.

بأسمائها /biʾasmāʾihā/ unique word that, once segmented, gives:
/bi/ "with"
/ʾasmāʾi/ "name, appellative"
/hā/ she
Translation: with her names (appellatives)

Information is cleaned up and enhanced through the filtering/normalization step: each token is analyzed and submitted for further processing. The actions usually performed to accomplish this task are (i) management of stop words, (ii) identification of stem, (iii) use of thesauri and spell-checking, and (iv) association of a score to the words (weighting). Unimportant terms and elements can be removed from indexing. The punctuation marks, articles, prepositions, and other text elements without specific value are included in this batch (management of stop words).[33]

The user often knows what to look for, but he does not know how to describe it formally. The stemming[34] techniques as well as the use of thesauri[35] tend to increase, as much as possible, the information conveyed by the original token. Textual resources (documents), after this step, grow in relevance and can be also retrieved by queries having similar key-words. A token derived from

[32] The 'regular expression' represents default practice and character families that identify common and uniform properties in textual resources.

[33] The conventions adopted for stop-words treatment are the common practices for tokenization. They follow the default rules adopted in the processing tools used.

[34] The term 'stem' indicates the textual unit common to similar words.

[35] The term 'thesaurus' indicates the possibility of replacing several similar terms with a single 'canonical' one using a vocabulary.

a word spelled incorrectly may undergo correction before being stored in the index (under certain conditions). The last action is justified by the user's unawareness of a possible orthographic mistake inside the text (errors from the original text or errors resulting from digital acquisition).

In the light of all this, it may be necessary to assign a score to evaluate the relevance of textual resources. The conclusions derived from the pioneers of the Information Retrieval (Luhn, Zipf, Sparck Jones, Salton) as well as some modern studies (led by the web[36] industries and by scientific researches)[37] highlight the importance of the weighting process.

The G2A platform does not experience problems related to "weight/score" assignment in view of the fact that partial matching functionalities[38] are much less necessary if compared to Boolean retrieval features,[39] since the requirements are born in a context where each data has a specificity known a priori (i.e. the information led through the text has never been obtained by way of probabilistic reasoning). The words in a text, generally, have different importance and we need tools to better identify the differences. The term frequency inside a document or inside an entire collection[40] can be evaluated and attained as a first weight criteria. From this result and taking into account the Zipf[41] law, we have a "scoring" system aimed at retrieving a set of more relevant and accurate data. The index (or indexes), afterwards, will have a key-term parameter from which one can obtain an accurate search and retrieve a balanced document set. Along with the full text[42] indexing system, the platform has developed components in order to upgrade raw textual data with extra information: additional elements of linguistic nature are combined to each processing unit (token). The text analysis step has been carried out in a semi-automatic way thanks to the morphological engines which are open source,[43] suitably customized,[44] for both the Greek and Arabic text. The base granularity is the word unit[45] and the data enrichment process produces a structured information stored in a look-up table.[46] Each token, hence, has a list of associated values and manages the links to the document collection (Tab. 4 shows an outline of this information).

---

[36] The most famous and monetized algorithm is the Page-Rank developed by Larry Page at Stanford University, and Google co-founder with Sergey Brin.

[37] There are many scientific studies related to relevance and weighting topic. An example is the vector space model, latent analysis (LSA), and its relative techniques (SVD, Random Indexing, etc.).

[38] The matching component, in an information retrieval system, is responsible for checking document search after a user survey. The partial matching indicates information retrieval methodologies based on statistical and similarity operations.

[39] The Boolean retrieval or exact matching is a series of techniques and methodologies where a search result is expressed by Boolean operations. They compare the search keys with those stored in the indexes to verify the resource presence or absence. The methods act with no hypothesis of similarity between the search keywords and the indexed terms.

[40] H.P. Luhn, "The automatic creation of literature abstracts", *IBM Journal of Research and Development* 2/2 (1958), p.159-65, [URL: <http://www.research.ibm.com/journal/rd/022/luhn.pdf>].

[41] Zipf's law describes the relation between the frequency of a term in a document or collection and its position (rank), according to a frequency order. The law shows that the multiplication between the two values (frequency * rank) tends to be constant. See G.K. Zipf, *Human Behavior and the Principle of Least-Effort*, Addison-Wesley, Cambridge 1949; and Id., *The Psycho-Biology of Languages*, Houghton-Mifflin, Boston 1935.

[42] Full text indexing refers the construction of an index by using the terms extracted from the source text.

[43] The term 'open-source' means free access to the source code of a software application as well as the agreement to some rules and criteria, such as the free re-distribution of the original program. Details on the open source world are on the Open Source Initiative (OSI) website at URL: <http://opensource.org/osd> (accessed on March 2013).

[44] The morphological engines from Perseus Project (for the Greek) and the Buckwalter morphological system (for the Arabic) have been customized by F. Boschetti and O. Nahli respectively.

[45] The fundamental unit of analysis is the token, or even the sub-token. This is clear for compound words.

[46] The look-up table is synonymous in this paper with physical or back-end index, where each key corresponds to a list of additional elements.

Tab. 4. Exemplification of the linguistic enrichment information on a pericope.

```
<?xml version="1.0" encoding="UTF-8"?>
<add>
        <field name="analysis_ar">
                <w prog="0" id="p03747" start="0" end="5" form="لأنّه" lemma="لِ أَنَّ هُ"
                        root="# أنن #" pos="Pr Pa Pn" voc="لأَنَّهُ" token="لأنّه" />
                <w prog="1" id="p03748" start="6" end="11" form="يشتاق" lemma="اِشْتَاقَ"
                        root="شوق" pos="Vb" voc="يَشْتَاق" token="يشتاق" />
                <w prog="2" id="p03749" start="12" end="15" form="إلى" lemma="إِلَى"
                        root="إلي" pos="Pr" voc="إِلى" token="إلى" />
                <w prog="3" id="p03750" start="16" end="21" form="الفعل" lemma="اَلْ فِعْل"
                        root="# فعل" pos="Pa No" voc="الْفِعْل" token="الفعل" />
                <w prog="4" id="p03751" start="22" end="28" form="كثيراً" lemma="كَثِيراً"
                        root="كثر" pos="Av" voc="كَثِيراً" token="كثيراً" />
                <w prog="5" id="p03752" start="29" end="33" form="وإلى" lemma="وَ إِلَى"
                        root="# ألي" pos="Cj Pr" voc="وَإِلى" token="وإلى" />
                <w prog="6" id="p03753" start="34" end="39" form="زَيْن" lemma="زَيْن"
                        root="زين" pos="No" voc="زَيْن" token="زَيْن" />
                <w prog="7" id="p03754" start="40" end="47" form="الأشياء" lemma="اَلْ شَيْء"
                        root="# شيأ" pos="Pa No" voc="الأشْيَاء" token="الأشياء" />
                <w prog="8" id="p03755" start="48" end="52" form="التي" lemma="اَلَّذِي"
                        root="ذا" pos="Pn" voc="التي" token="التي" />
                <w prog="9" id="p03756" start="53" end="57" form="رآها" lemma="رَأى هَا"
                        root="رأي #" pos="Vb Pn" voc="رَآهَا" token="رآها" />
                <w prog="10" id="p03757" start="58" end="60" form="في" lemma="فِي"
                        root="في" pos="Pr" voc="في" token="في" />
                <w prog="11" id="p03758" start="61" end="67" form="العقل" lemma="اَلْ عَقْل"
                        root="# عقل" pos="Pa No" voc="الْعَقْلِ" token="العقل" />
        </field>
        <field name="analysis_gr">
                <w prog="0" id="129063" bibref="4.7.13.6" form="καὶ" lemma="καί"
                        pos="conj" uppercaseform="ΚΑΙ" start="0" end="3" token="καὶ" />
                <w prog="1" id="129064" bibref="4.7.13.6" form="κοσμεῖν" lemma="κοσμέω"
                        pos="verb" uppercaseform="ΚΟΣΜΕΙΝ" start="4" end="11" token="κοσμεῖν" />
                <w prog="2" id="129065" bibref="4.7.13.6" form="ὀρεγόμενον" lemma="ὀρέγομαι"
                        pos="participle" uppercaseform="ΟΡΕΓΟΜΕΝΟΝ" start="12" end="22"
                        token="ὀρεγόμενον" />
                <w prog="3" id="129066" bibref="4.7.13.6" form="καθὰ" lemma="καθά καθ’ἅ"
                        pos="adv" uppercaseform="ΚΑΘΑ" start="23" end="27" token="καθὰ" />
                <w prog="4" id="129067" bibref="4.7.13.6" form="ἐν" lemma="ἐν" pos="prep"
                        uppercaseform="ΕΝ" start="28" end="30" token="ἐν" />
                <w prog="5" id="129068" bibref="4.7.13.6" form="νῷ" lemma="νοῦς"
                        pos="noun" uppercaseform="ΝΩ" start="31" end="33" token="νῷ" />
                <w prog="6" id="129069" bibref="4.7.13.6" form="εἶδεν" lemma="εἶδον οἶδα"
                        pos="verb" uppercaseform="ΕΙΔΕΝ" start="34" end="40" token="εἶδεν," />
        </field>
</add>
```

The Arabic segment which this linguistic analysis refers to is located at I, p. 19.3-4 Badawī, while the Greek one is located at Plot., *Enn.* IV 7 [2], 13.6.

The points below list the steps in order to produce and handle data:

- Extraction of tokens from the text
- Indication of the place where the term occurs
- Transformation of Greek words in capital letters[47]
- Extending Arabic words by vocalization
- Assignment of grammar categories to each word (morpho-syntactic analysis)[48]
- Lemmatization of the Greek and Arabic word-forms
- Attribution of the morphological root to the Arabic words
- Mark-up the polyrematics/agglutinated forms (if any)
- Attribution of the graphic variants to the tokens (if any)

Table 6 shows the definition of the back-end index attributes, which are populated by the analysis phase.

The schema[49] is inspired by the common representation of information known as ConLL data format, and it is largely adopted in natural language processing field. The index has a number of attributes that reflect the points mentioned above:

a) the *Token* field represents the processing unit extracted from the text, on which are grounded all the subsequent steps of the analysis;

b) the *Pericope* field consists of the identification number (ID) of the pericope which the token belongs to;

c) the *Offset* field expresses the position of the token within the pericope. Thanks to this data it is possible to take into account the proximity relations;

d) the *Lang* field refers to the language or alphabet of the token;

e) the *Status* field defines additional information: for example, the token could be part of a polyrematic term or express agglutinative phenomena. Terms with graphic variants or tokens part of a hyphenation term could be managed through the status field;

f) the *Normalization* field provides the possibility of transforming and/or harmonizing the tokens, e.g., returning the word in uppercase form (as in Greek) or eliminating the sign of tatweel[50] (عملي / عمـــلـــي, as in Arabic);

g) The *Extension* field is left free for any future customization, such as the management of the variant readings;

h) The fields *PoS*, *Lemma*, and *Root* specify the morpho-syntactic information consequence of linguistic analysis.

---

[47] The operation of transforming Greek words in capital letters is justified by the fact that neutralizing Greek diacritics as breathing marks and accents the search engine can construct a list of documents (a. k. a. result. set) settled with more data; as counterpart, however, the result set presents also data which are not pertinent to the user query (see below, p. 226). For example, if the index handles only the word form ἡ, users are not able to retrieve the contexts where the word form ἥ appears; on the other hand, if the index handles the word form capitalized H, users are able to retrieve contexts where also the word forms ἡ, ἥ, ἧ are present.

[48] For a complete linguistic description of the Arabic functions and characteristics, please refer to the contribution by O. Nahli, in this volume.

[49] The term 'schema' indicates the elements definition of a data structure.

[50] A correct analysis of a text should include the harmonization of the spelling forms. As an example, let us consider the well-known encoding problem of tatweel (also called 'kashida'). The encoding of this character is described in the Unicode standard (Version 6.2, cf. URL: <http://www.uni- code.org/charts/PDF/U0600.pdf>) and it is based on the standard ISO labeled "ISO/IEC 8859-6:1999, 8-bit single-byte coded graphic sets –Part 6: Latin/Arabic alphabet" (see the mapping file at URL: <ftp://ftp.unicode.org/Public/MAPPINGS/ ISO8859/8859-6.TXT>). The tatweel sign is used only for graphic and typographic purposes in order to stretch the characters in a word, and it should be removed from the token.

Some fields may take multiple values, for example in the case of token compounds, the elements have sub-tokenization entities[51] (Tab. 5).

Tab. 5. Exemplification of sub-tokenization elements.

TOKEN: وَلِتُدَبِّرُهَا walitudab~irohA VERB_IMPERFECT
SUB-TOKEN: wa / li / tu / dab~ir / o/ hA
ANALYSIS: wa=CONJ / li=SUBJUNC / tu=IV3FS / dab~ir=VERB_IMPERFECT / o=IVSUFF_
MOOD:JS hA=IVSUFF_DO:3FS+

Tab. 6. Back-end data index.

Token	Lemma	PoS	Root	Norm	Peric	OffSet	Lang	Ext	Status

An index rich in information (textual and linguistic) makes it possible to carry out techniques[52] for better retrieving documents as an accurate result of the query. However, the discussion of these topics goes beyond the scope of this paper.

3. Construction of indexes

Second level data structures, called 'inverted indexes',[53] are generated for each significant phenomenon. Indexes provide flexibility as well accuracy in the information retrieval and analysis activities. These kind of structures are constituted by a list of couples <k,d> (two-dimensional vectors) where 'k' stands for the 'key-term'[54] and 'd' stands for a list of references to the text. The index could hold statistical parameters, such as the term frequency of a single document or of the whole collection/corpus. The key-terms indexed give linguistic access to resources; they also express other nature categories as could be named entities or domain terms. A relevant outcome of this work is the possibility of performing combined advanced searches, i.e. to find text contexts within the scope of a language while applying restrictions on the parallel one. This special approach provides a method for studying complementarity or links among the texts. Lemmatization and grammatical information simplify the understanding and, consequently, give the possibility for a better text reading.

4. Storage and Information Management

Software systems and the technologies involved are part of the open-source world and are standardized by universally recognized organizations. They can, therefore, be used without any economical and copy-right limitations. They are:

---

[51] A comprehensive explanation of grammar categories (parts of speech) used by the Arabic morphological engine is accessible at URL: <http://www.nongnu.org/aramorph/english/grammatical_categories.html> (accessed on March 2013).

[52] Such techniques are called 'extended query' and 'extended index'.

[53] The inverted index consists of a set of records containing the word wanted, and a sequence of pointers to information concerned to it. The word 'inverted' explains the reversal of the research direction: first the key and then the document containing the key.

[54] The key is the term by which machines and users are able to achieve data without spanning through either the entire collection or the entire index.

- Java enterprise platform (related to the *JSR-316* specification), which concerns the technological environment chosen for software developing;
- *Apache Lucene*, which is the software library for text indexing used, whitin the context of G2A, to parse Arabic and Greek documents;
- *eXist-db* platform, which is the XML native environment able to manage semi-structured databases;
- *TreeMap* data structures, which are used for efficient and flexible navigations of all front-end indexes.

Java Enterprise Edition (JEE6) allows software developers to divide the system into three main areas:

a) presentation and Web interface (View);
b) logic application and business level (Control);
c) persistence of the domain entities (Model).

This type of division follows a well-known architectural pattern[55] called Model-View-Controller (MVC)[56].

Lucene is an open-source library widely used by the community of developers for document indexing systems. It is recognized as one of the most efficient and flexible solutions for full-text search operations. Lucene is able to index textual resources regardless of their nature: they can be a database, a non-structured file, a formatted file, a pdf file, or other kinds of data. The available features allow the user to create catalogs, to build archives, and to conduct advanced search operations on acquired plain texts. The aforementioned software library is composed of three basic elements:

- the *index*, represented by a structure referring to all the analyzed documents;
- the *document*, which is an internal structured representation of the textual source;
- the *field*, i.e., an element of the aforementioned document, consisting of a pair 'name-content'. The field manages the token stream to be indexed.

The user[57] has the possibility to perform complex search operations through a complete binary/Boolean[58] syntax, including wildcards,[59] searches[60] in range, and fuzzy operations.[61]

The parallel text segments and their linguistic analyses were encoded using the XML markup language. An analysis of state-of-the-art XML-based applications and the relative adopted persistence strategies (see, for example, the website *history.state.gov*) suggested the use of *eXist-db*, an efficient resource management system for structured data (another possible choice was *BaseX*).[62] Exist-db

---

[55] In software engineering, a pattern (or design pattern) can be defined as a general design solution to a recurring problem. See E. Gamma - R. Helm - R. Johnson - J. Vlissides (eds.), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Boston 1995.

[56] Cf. S. Burbeck, "Applications Programming in Smalltalk-80TM: How to use Model-View-Controller (MVC)," 1992, URL: <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html> (accessed on March 2013).

[57] An overview of the use cases is part the contribution by S. Marchi, in this volume, "Part II: towards a user manual".

[58] Using the Boolean Algebra operators it is possible to perform more advanced searches. See "Conclusions", for more details.

[59] Wildcards are special characters representing, within a string, other sequences of characters.

[60] A range search allows defining proximity parameters among the query terms.

[61] A fuzzy search differs from the Boolean search since the operations of inclusion and/or exclusion are based on probability. The documents are evaluated on similarity measures.

[62] The BaseX project is accessible at URL: <http://basex.org> (accessed March 2013).

performs queries on archived contents and offers useful features to create, read, modify, and delete resources.

To date, the system allows the user to get direct access to the text structure and is able to extract information without any internal conversion or data model adaptation. The intended feature is to query data directly using the DOM[63] standard technology. XPath[64] and XQuery[65] are the main technologies directly involved in the construction of queries. Exist-db is completely free for academic and community use; it integrates seamlessly into the Java development environment and it incorporates specific modules to work with the Lucene indexing library. The TreeMap is a data structure optimized for internal data manipulation and front-end index sorting. As implied by the name, a TreeMap manages information using a key-value mechanism (Map) on data represented as a balanced binary tree.[66] The Java programming language creates objects of TreeMap type which are constituted of:

- a *key* representing the stored term;
- a *value* associated to the key.

For example, in Tab. 7 and in Tab. 8, the word *String* stands for the term and the word *Integer* is a number expressing the term frequency, evaluated taking into account the whole corpus. The TreeMap data structure makes it possible to explicitly state the order of terms and read all the tree elements in a fixed way. This solution is essential for the G2A platform as it deals with two distinct lexicographic indexes (Greek and Arabic).

Tab. 7. Object TreeMap for the Arabic index (Java).

```
new TreeMap<String, Integer>(
                    Collator.getInstance(new ULocale("ar"))
                    );
```

Tab. 8. TreeMap object for the Greek index (Java).

```
new TreeMap<String, Integer>(
                    Collator.getInstance(new ULocale("grc"))
                    );
```

For the sake of completeness, we briefly show in what follows the typical evaluation parameters used for indexing and for information retrieval systems.

---

[63] An encoded XML document has a tree structure. It is composed of elements such as "root", "child", "father", etc. This structure is also known as DOM (Document Object Model) representation.

[64] XPath is a language used to address elements of the XML document. It is based on a very expressive syntax thanks to the so called 'location steps'. A typical XPath statement can be: collection ('/db/pericopes')//doc//w[@lemma eq 'locus'].

[65] XQuery is built using XPath. It is a flexible language used for querying and transforming XML resources. Expressions and queries are called FLWOR, an acronym that means *For - Let - Where - Order by - Return*.

[66] In computer science, a balanced binary tree is a structure conceived to provide very efficient construction and maintenance of ordered lists. It consists of elements (called nodes) linked in hierarchical parent-child relations. A data structure can be considered a balanced binary tree if: (a) each node has maximum two 'children', each of which may be the root of two other sub-trees (left and right); (b) given a node, values of nodes belong to its left sub-tree is not be greater than the right sub-tree values and vice versa; (c) given a node, the relative sub-trees have, approximately, the same number of nodes.

Search speed. Systems make extensive use of scalable and robust algorithms and technologies. Without going into quantitative details, we just mention the fact that the inverted indexes, built using binary trees, are the most effective data structures, at the state of the art, concerning search and sorting performance.

Recall. The ability to retrieve relevant information from a document collection is defined by a parameter called 'recall'. It is defined as the ratio between the number of relevant documents retrieved and the total number of relevant elements present in the collection (Fig. 3).

Precision. The ratio between the number of relevant documents retrieved and the total number of documents retrieved is called 'precision'.

Exhaustiveness. It is defined as the ability to index the greatest number of phenomena present in the text, para-text and extra-text of the digitized and stored documents. It may be related to the number of key-terms and meta-data assigned to each document.[67]

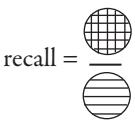
Specificity. It is the precision and accuracy of an index in describing a document and/or a topic. Specificity can also be seen as a measure of the number of documents related to a specific term or key.[68]

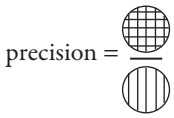
Fig. 3. Graphical exemplification of the relation between precision and recall.

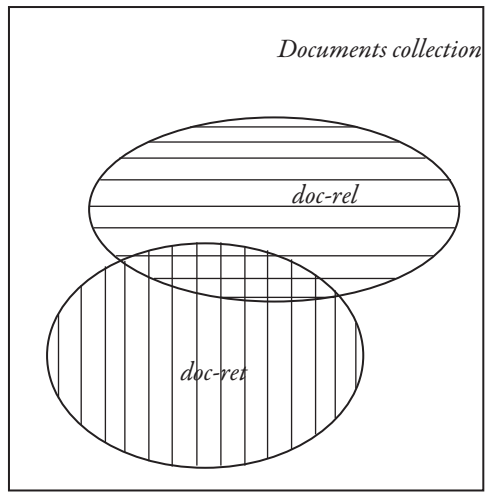
$$\text{precision} = \frac{|\text{doc-ret} \cap \text{doc-rel}|}{|\text{doc-ret}|}$$

$$\text{recall} = \frac{|\text{doc-ret} \cap \text{doc-rel}|}{|\text{doc-rel}|}$$

Doc-ret = document retrived
(documents obtrained after the query)
Doc-rel = document related
(effective documents looked for)







The parameters of recall and precision are often evaluated together (combining this two values we obtain other parameters, such as accuracy or F-measure). Typically, the greater the recall, the smaller is the accuracy of the results. In fact, the greater is the effort to retrieve all the relevant documents, the higher will be the probability that, in output, there is a lot of irrelevant documents, thus decreasing the accuracy. Conversely, the greater the precision, the smaller is the recall of the result set. In other words, to avoid retrieving irrelevant documents some relevant documents can be overlooked.

The G2A search engine returns all the information the user has asked for, and only it, since all the information stored in the system has been structured and indexed. Hence, the precision and the recall

[67] K.S. Jones, "Statistical interpretation of terms specificity and its application in retrieval", *Journal of Documentation* 60/5 (2004), p. 493-502 (DOI:10.1108/eb026526).

[68] Jones, "Statistical interpretation".

parameters are supposed to be equal to the theoretical maximum value (meaning that concerning the whole archive the set of retrieved documents is equal to the set of relevant documents). On the other hand, in the G2A system, execution speed and information exhaustivity constitute the relevant parameters.

G2A Variant readings component

Computational philology is a multidisciplinary field where computer science[69] and computer engineering[70] create synergies with the study of texts. The component assists the philologist in the investigation of ancient, modern or contemporary documents.[71] The domain and the basic functionality of a system in computational philology should take into account the tradition (witnesses) and the reconstruction of an original text (archetype/*Urtext*). The machine obviously cannot replace the scholar: the tool has the main purpose to facilitate the management, usability, production, and research within the context of the editorial work.[72] Similar systems have been proposed over the years [cf. HyperNietzsche, Anastasia, Collate-CollateX, SDPublish, Bambi, FAD, Version Machine, Juxta, TUSTEP]. Many of them are based on hyperlinks approach and automatic handling of data, or, anyhow, set up on previously encoded information.

Within the context of the project *Greek into Arabic*, a module of textual criticism has been designed, which is meant to assist the production of a critical edition. The main principle at base of the general and theoretical method is such to manage the resources (witnesses of the tradition which can be manuscripts and printed editions) as scanned images (facsimile) and to promote the complete transcription of the witness considered "the best choice".[73] Therefore, the scholar draws up the issue through annotations and comments on portions of images that are being analyzed. The scans of witnesses and the resource chosen as the "basis of collation"[74] can be analyzed in parallel and mutually studied. The reconstruction of the text on the basis of the variant readings of other sources can be automatically rebuilt thanks to the critical apparatus, which is positive. In this way, the formal elements to achieve algorithms automatically managed would not diminish the role of the scholar experience in order to establish the text.

To complete the necessary domain overview, which the digital and computational philology is based on, it is extremely important to highlight the management of the variant readings offered to the scholar's choice. These latter (variant readings) are present in the manuscripts as the whole text tradition (primary sources)[75]. Domain entities have been identified and outlined in a diagram

---

[69] Computer sciences deal with natural language processing, problem solving, algorithm optimization, and artificial intelligence.

[70] Computer engineering deals with complex systems design, multiple architectures, accessibility and usability, and studies integrations and protocols of communication among distributed services.

[71] A. Bozzi, "Edizione elettronica e filologia computazionale", in A. Stussi (ed.), *Fondamenti di critica testuale*, Il Mulino Manuali, Bologna 2006, p. 207-32.

[72] M.S. Corradini, "Formalisation des variantes à des fins computationelles: vérification de l'hypothèse expérimentale sur un texte occitan", in D. Billy - A. Buckley (eds.), *Études de langue et de littérature médiévales offertes à Peter T. Ricketts*, Turnhout 2005, p. 355-68.

[73] Bozzi, "Edizione elettronica e filologia computazionale", p. 216-19. The purpose of this section is to describe, in a very concise way, the theoretical model behind the technical design of the textual critical component; for more information, see A. Bozzi's contribution in this volume.

[74] The base of collation is a linear transcription of a single source, chosen following the Bédier principle (the so-called "bon manuscrit").

[75] An overview on the primary and secondary sources is part of the contributions of A. Bozzi and F. Boschetti in this volume.

(not yet exhaustive of the overall complexity of the philological domain). It constitutes a first conceptual model of the problem; nevertheless, it offers a basic objects design of the textual criticism component (Fig. 4).

Fig. 4. First draft of a computational philology model (object oriented).

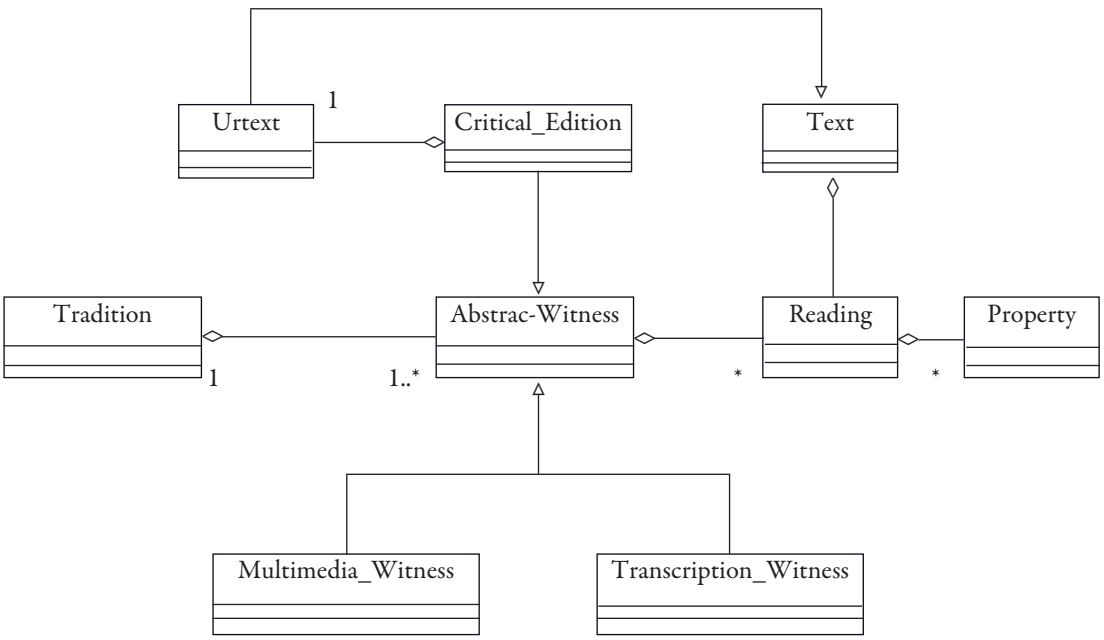


Figure 5 shows the web interface of the designed textual criticism module and the related scenario made possible by the system. The entities (objects in computer representation) interact with scholars through an interface, which has to be graphical and user-friendly.

The philological component displays the selected document as a basis of collation (resource on the left, Fig. 5). The facsimile (scan) must be transcribed or, if an external standard formatted transcription is available, it can be imported and referred to the image (the transcription is available selecting the label "transcription" placed nearby the label "multimedia resource" as Figure 5 shows in the upper area reserved for the collation base view, in the left hand side of the graphical schema).

In the right hand side of the interface we find an area dedicated to witnesses (as a whole), from which it is possible to record the variant readings. The scholar can manage them facing the text transmitted from different witnesses. He might select portions of text through interactive tools and notes. The informative content of the reading is automatically recorded and displayed at the bottom part of the workspace which is provided for the critical apparatus management and maintenance. It is possible to associate a typology and a literary comment to each reading.[76] A label to each variant is

[76] Bozzi, "Edizione elettronica e filologia computazionale".

gathered to generate useful indexes and have a synoptic and global overview.[77] The system gives the user the choice of submitting his own reading, which features in a specific area, as if it were a variant reading of the manuscript tradition.[78]

Fig. 5. Graphical User Interface (GUI) Designed for the variant reading scenario.
The system stores, in a specific area of the recording interface, the information about the readings: "locus" means the parameters referred to the page, line and position of the selected characters. The system stores the coordinates of the selection area if the scholar marks the scanned source. The infrastructure developed on the groove of the model introduced above is able to manage graphically concordances of all the stated and recorded word-forms.

Conclusions

Obtaining a refined and advanced parallel search in an efficient way is one of the task of G2A Web application. This is made possible thanks to the text processing implemented within this project. Analyses and indexing are focused on some book of the *Enneads* of Plotinus (fourth, fifth and part of the sixth) and the pseudo-*Theology* attributed to Aristotle.

Scholars, through G2A platform, are able to perform queries via a Web graphical user interface (GUI) based on significant linguistic parameters such as:

(1) the linguistic difference between a term and another;
(2) the position in which a term appears;

---

[77] For detailed analyses and a case study see Corradini, "Formalisation des variantes", where it is possible to evaluate the method proposed, in particular on variants and the *stemma codicum* reconstruction.

[78] The model, as described by A. Bozzi in this volume, handles variant readings and editorial conjectures in same way. This approach has been tested by some pilot projects and prototypes developed at the ILC under the direction of A. Bozzi. See A. Bozzi, "Towards a Philological Workstation", *Revue informatique et statistique dans les Sciences humaines* 29 (1993), p. 33-49. and Corradini, "Formalisation des variantes".

(3) the frequency which a term appears in a document;
(4) the status of the searched term.

The final purpose of the system is the establishment of a cyber-infrastructure, collaborative and flexible, able to assist the philologist/philosopher in his research and textual criticism activities, such as the analysis of the variant readings or the production of the *textus constitutus*.

To reach this goal, a computational philological module was formalized and designed. The development of the software components is in progress within the ILC software laboratories working at the ERC ADG 249431 *Greek into Arabic. Philosophical Concepts and Linguistic Bridges*.

Finito di stampare nel mese di settembre 2013
presso le Industrie Grafiche della Pacini Editore S.p.A.
Via A. Gherardesca • 56121 Ospedaletto • Pisa
Tel. 050 313011 • Fax 050 3130300
www.pacinieditore. it