Co-funded by the
Erasmus+ Programme
of the European Union

EULALIA

European Latin Linguistic Assessment

Erasmus+ Strategic Partnership for Higher Education (2019-2022)

(2019-1-IT02-KA203-062286)

https://site.unibo.it/eulalia/en

O1: European Latin Language Certification – Basic Level

Methodological and Pedagogical tools

LEXICON

(English Version: 31.05.2021)

Project Coordinator:

Alma Mater Studiorum – University of Bologna (Italy)

Project Partners:

University of Köln (Germany)

Catholic University of the Sacred Heart – Milan (Italy)

University of Rouen (France)

University of Salamanca (Spain)

University of Uppsala (Sweden)

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

A New Latin “Core Lexicon”

Guido Milanese[*]

1 State of the art

The project of a standard Linguistic Assessment, as designed by EULALIA, requires a core lexicon dictionary. The offer in this field is wide: see e.g. the highly refined tools offered by the Perseus Project[1] or the list provided by the Dickinson College (team led by Christopher Francese)[2]. However, the Dickinson College list is limited to 1000 words, while the EULALIA programme requires a larger amount of words for higher levels. Moreover, these list do not inform on the sources being used for dressing up the lists; and it was agreed by the members of the EULALIA committee to provide potential students with a list prepared using standard Classical Latin texts, i.e. the texts that are part of the real “canon” used in European schools.

For this reasons, the old and venerable *Vocabulaire de base du latin*, published in 1984 by a Besançon team (A.R.E.L.A.B.) that made use of the *Dictionnaire fréquentiel* published by the Liège LASLA group in 1981, can still be used as a reliable “basis” to build a new core vocabulary suitable for the needs of the teaching of latin in European schools and universities. The advantages of this work are:

1. amount of words: 1600 words are compatible with the EULALIA programme;
2. a clear list of the Latin texts used to prepare the lexicon:

   «Le corpus analysé dans cet ouvrage comprend, en partie ou en totalité, les œvres de Catulle, César, Cicéron, Horace, Juvénal, Ovide, Perse, Properce, Quinte-Curce, Salluste, Sénèque, Tacite, Tibulle,

[*] EULALIA – European Latin Linguistic Assessment

[1] `https://www.perseus.tufts.edu/hopper/help/vocab`

[2] `http://dcc.dickinson.edu/vocab/core-vocabulary`

Tite-Live, Virgile, Vitruve»[3]

No list is perfect, of course, and the lack of authors such as Terence or Lucretius can be lamentable, but in order to prepare a "neutral" list of word of the Late Republican and Early Imperial Latin the choice of authors is adequate. The present writer believes that Mediaeval Latin can provide teachers with a large amount of works very effective for the teaching of Latin, but also in this case the core lexicon can be easily integrated.

2 Scan and OCR of the Besançon lexicon

The French–Belgian lexicon was published as a book[4]. Unfortunately, the quality of the original print is that of a book printed in the '80 using a camera ready prepared with the devices of the time. The list of words was scanned using a standard HP OfficeJet Pro 7720, and the OCR was performed by the well known OCR programme `tesseract`, with the option `latin`[5]. The list was corrected manually but, to my surprise, even the first `tesseract` output was already very good.

3 Morphological tagging

The final list provided by the Liège–Besançon lexicon is a bare list of words and frequencies, with some notes added here and there. In order to use this list for a real "core lexicon" list, it was obviously necessary to add a suitable morphological tagging. For this purpose, I used `treetagger`, the morphological programme designed by Helmut Schmid in the TC project at the Institute for Computational Linguistics of the University of Stuttgart and now hosted by the University of Munich, Germany[6]. Running `treetagger` on a list of words is an error-prone approach: the parser cannot perform a contextual analysis, and

[3] G. Cauquil and J.Y. Guillaumin, *Vocabulaire de base du latin (alphabétique, fréquentiel, étymologique)*, ARELAB, Besançon 1984, p. 4.

[4] The bare list of words was reprinted in more recent works, such as Paolo Lamagna, *Il lessico latino di base*, Bompiani, Milano 1999.

[5] About `tesseract`, see Guido Milanese, *Filologia, letteratura, computer. Idee e strumenti per l'informatica umanistica*, Vita e Pensiero, Milano 2020, pp. 103-107, and the online documentation at `https://github.com/tesseract-ocr`. The programme is FOSS (Free and Open Source Software) and was run on a Linux Mint 20.1 system.

[6] See `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`, and `https://www.cis.lmu.de/~schmid/` for recent bibliography.

entries e.g. as *quam* are obviously opaque[7]. The morphological analysis offered by `treetagger` was manually revised; about 15% of the tags were wrong, as expected. The ARELAB notes were used to correct the `treetagger` tag, whenever different from the morphological category offered by the ARELAB note.

4 The problem of function words

During a meeting of the EULALIA team, Mélanie Lucciano (Université de Rouen-Normandie) noticed that a "pure" frequency list was not suitable to be used as a tool for the teaching of a language because function words ("mots-outils", "morfemi grammaticali", "parole vuote"...) are very common in any language and are listed in the highest rank of a frequency list. As a consequence, if we assume a list of 400 words for an **A1** level, students would learn a great amount of conjunctions and prepositions, at the expense of verbs, nouns, and so on.

The morphology classes listed by `treetagger` are the following ones:

- ADJ
- ADJ:NUM
- ADV
- CC
- CS
- DET
- DIMOS
- ENCL
- ESSE
- EXCL
- FW
- INDEF
- INT
- N
- POSS
- PREP
- PRON
- REL
- V

Since we are dealing with the teaching of the language, and the purpose of adapting the original ARELAB wordlist is to avoid too many function words, I would

[7] No real difference using the more recent creation of Schmid, called 'RNNTagger': see `https://www.cis.lmu.de/~schmid/tools/RNNTagger/`.

propose to consider as "function words" the following categories: conjunctions, prepositions, and clitics.

A simple SNOBOL4 program run on the ARELAB list outputs these results[8]:

- on the whole list (1635 words) "function words" are 78, i.e. 4.1260 %
- on the first 600 words, "function words" are 55, i.e. 9.100 %
- on the first 400 words, "function words" are 45, i.e. 11.100 %

Besides that, the 57% of the "function words" is listed among the first 400 words, and almost the 70% among the first 600 words. The consequence is that the list of the most frequent words is not suitable to test the linguistic competence of students, because among the first 400/600 positions there are too many function words. The purpose of this revised list is to balance function words and "semantic words", moving some semantic words from the lower range of the list (= words after n. 400 and n. 600) in order to obtain a more balanced percentage of semantic/function words. In other words, the first 400 and 600 words should feature a percentage of function words similar to the one of the full list.

The list was revised moving some function words to the lower positions of the list, making sure, however, that essential function words (such as *quia* or *si*) be listed in the lists for beginners. The work was done with a computer-assisted approach, i.e. checking the balance of words step by step.

The revised list features now the following percentages:

- on the first 600 words, "function words" are 46, i.e. 7.4 %
- on the first 400 words, "function words" are 30, i.e. 7.2 %

Even if these percentages do not mirror exactly the data of the complete list, this seems to me a reasonable compromise – although questionable, of course, and open to further improvements – between exactness and practical teaching needs.

5 Final output

The final output was produced in the form of a `csv` (Comma Separated Values) list, suitable to be read by standard programmes such as LibreOffice *Calc* or MSOffice *Excel*. I added a simple header:

[8] Although a "niche language", `snobol4` is well maintained, particularly thanks to the efforts of Phil Budne: see `https://www.snobol4.org/`. The most recent version of the language was made available in December 2020. On this language, particularly suitable for text analysis, see Milanese, *Filologia, letteratura, computer* cit., pp. 241-244. `Snobol4` runs on Linux, Windows, and OSX.

N , VOX , MORPHO , NOTAE , FREQ

where **N** = number, **VOX** = the word listed, **MORPHO** = the morphological tage, **NOTAE** = the original note of the Liège–Besançon *Vocabulaire*, and **FREQ** the frequency of the word. For example:

```
16,QUIS,PRON,interr.,4555
```

The information added by the *Vocabulaire* is added to specify the kind of pronoun *quis* belongs to. Whenever the information provided by the *Vocabulaire* is the same of the morphological tag, it is silently omitted.

To make the list more readable, a PDF file was produced from the CSV list, using `pandoc`[9].

6 Lists from the list

Since the purpose of all of this is providing teachers and students with convenient, easy to use tools, I rearranged the lists in various forms. The first and obvious step was to print the lists of the first 400 and 600 words.

6.1 Alphabetical lists

All the lists (i.e. the whole list and the "400" and "600" lists) were sorted in alphabetical order, in order for the teacher or the student to locate a given word browsing the list alphabetically. The complete alphabetical list was also transformed into a PDF file.

6.2 Morphological lists

All the lists (i.e. the whole list and the "400" and "600" lists) were sorted in morphological order, in order for the teacher or the student to locate all the words of a given morphological category. The complete morphological list was also transformed into a PDF file.

7 A GUI of the list

A very simple graphical user interface, to make the use of these lists easier, has been developed by the present author and is available to beta testers interested in this project.

[9] See `https://pandoc.org`.

8 Code of the snobol4 and bash scripts

8.1 The snobol4 script

The snobol4 script for the calculus of percentages:

```
    P_type = break(',') len(1) break(',') len(1) (break(','). M_type)
    KL =0
    KE =0
loop    Line = input    :f(loop_end)
    KL = KL +1
    Line ? P_type
    ((leq(M_type,"CC")), (leq(M_type,"CS")), (leq(M_type,"ENCL")),
+  (leq(M_type,"PREP")))  (KE = KE +1) :(loop)
loop_end  Output = KE'/' KL
    Extreme = KE *100
    Percentage = Extreme / KL
    Modulus = Remdr(Extreme,KL)
    Terminal = Percentage'.' Modulus
end
```

The first line declares a pattern: in a given CSV line, the cursor moves to the first comma, then one character, repeats the same steps, and at the 3rd attempt saves the information as morphological type. Then we initialise the counter of lines (KL) and the counter of empty words (KE) at zero[10]. The loop reads all the lines from standard output; if the morphological type is an empty word the KE counter is increased by 1. After the loop, some simple math operations produce the desired output.

The same programme is run on the first 400 / 600 words adding a limit:

```
    lt(KL,601)
```

In this case, the loop will end at the 600th line (`lt` = less than).

8.2 The bash scripts

The bash scripts used to arrange the lists in alphabetical or morphological order are simple `sort` script, as the following one:

```
sort -t, -k3,3 -k2 frequentia-rev-1-400.csv\
> frequentia-rev-1-400-morph.csv
```

[10] This step is not necessary, but data initialisation makes the programme easier to read and to understand.

This instruction sorts the “400” list on the 3rd field (the morphological type) and then on the 2nd one (= alphabetically), writing the output to a new file (`frequentia-rev-1-400-morph.csv`). A comma is used to separate fields (“`-t,`” or “`-t','`”).

8.3 From CSV to PDF using pandoc

Pandoc is a very efficient command line programme able to translate from many currently used formats (e.g. DOCX, LaTeX, markdown) to other formats[11]. The most recent version of `pandoc` can read a CSV file and transform it to other formats, including PDF (calling silently TeX).

This is the simple instruction:

```
pandoc -f csv frequentia-format-rev-morph.csv\
 -s -o frequentia-format-rev-morph.pdf
```

Pandoc reads a CSV file (`-f` means “from”) and outputs a PDF file. The same scheme applies to all the other files.

References

Cauquil, G. and J.Y. Guillaumin, *Vocabulaire de base du latin (alphabétique, fréquentiel, étymologique)*, ARELAB, Besançon 1984.

Delatte, L., Et. Evrard, S. Govaerts, and J. Denooz, *Dictionnaire fréquentiel et index inverse de la langue latine*, L.A.S.L.A. (Laboratoire d’analyse statistique des langues anciennes), Université de Liège, Liège 1981.

Lamagna, Paolo, *Il lessico latino di base*, Bompiani, Milano 1999.

Milanese, Guido, *Filologia, letteratura, computer. Idee e strumenti per l’informatica umanistica*, Vita e Pensiero, Milano 2020.

[11] See `https://pandoc.org/`.