

# insertLemmas.py

Michael Percillier

## 1 Instructions

- In the Terminal, change the working directory to the current directory using the `cd` command
- Run the script with `python insertLemmas.py corpusDir`, where `corpusDir` stands for the directory containing your corpus files with the file ending `*.psd`
- When running with Python 3, use the script `insertLemmas3.py` instead
- When running the script on the PLAEME corpus, use the flag `-p` or `--plaeme`, so `python insertLemmas.py -p corpusDir`
- Further optional flags are available for the PPCME2 only:
  - Generate a report detailing how many verbs were annotated in each text. To do so, rename corpus files to include the period (m1-m4) and the textual base ('en': Middle English original text, 'fr': text translated from French), e.g. `cmaelr3.m23.psd` to `m2-en.cmaelr3.m23.psd`. Once the files have been renamed, you can run the command as `python insertLemmas.py corpusDir -s` or `python insertLemmas.py corpusDir --summary`
  - Insert manuscript and composition dates in the file names with the flag `-d` or `--dating`

## 2 Details

The script `insertLemmas.py` inserts lemma information for lexical verbs in the PPCME2, the PCMEP, or the PLAEME, relying on three files:

1. `lemmas.csv`: a CSV table of form-lemma links
2. `frenchMEDverbs.csv`: a list of verbs copied from French between 1066 and 1500, provided by the OED. If a verb occurs in this list, its etymology is marked as `e=french`, and as `e=nonfrench` otherwise. Some gaps in the original list have been added to the file `frenchMEDverbs.csv`, and marked with the comment “verb not in OED list” if a French-based verb did not occur in the OED list at all, or “link gap” if the verb occurred in the OED list, but could not be traced to the MED entry due to a missing link between the two resources. Additions can be made directly to

`frenchMEDverbs.csv`. Please add the appropriate comment when doing so.

3. `textDates.csv`: a table containing the dates for each text, to be inserted in the file names of the lemmatised corpus (PPCME2 only)

## 3 Version history

### 3.1 Version 15 (2021-12-06)

- Updated list of French-based verbs

### 3.2 Version 14 (2021-10-06)

- Updated URL of BASICS Toolkit in the disclaimer
- Updated list of French-based verbs

### 3.3 Version 13 (2020-09-17)

- Added insertion of a disclaimer in the output files so that end users are made aware of the fact that their corpus files have been treated by the script, with links to further information
- Fixed a minor compatibility issue between Python 2 and Python 3 related to the display of credits when running the script

### 3.4 Version 12 (2020-07-17)

- Minor corrections and additions to `lemmas.csv`, based on verification of new lemmas encountered in the lemmatisation of the additional corpora PCMEP and PLAEME
- Added conversion of comments other than “doubt” as warnings:
  - forms wrongly tagged as verbs so that their actual word class appears as a `w` parameter, e.g. `@w=noun`, e.g. `@w=adjective`
  - or verb forms part of a compound as `@w=compound`,
  - or disagreement with the proposed lexe in PLAEME

### 3.5 Version 11 (2019-11-05)

- Minor corrections and additions to `lemmas.csv`

### 3.6 Version 10 (2019-10-22)

- Updated `lemmas.csv` to include forms that were previously not lemmatised in PCMEP and PLAEME
- Fixed an error that prevented PLAEME forms containing a  $\hat{\phantom{x}}$  superscript marker from having their lemma information inserted

- Added a Python 3 version

### 3.7 Version 9 (2019-04-30)

- Minor fixes to make the script compatible with other corpora, in this case the PCMEP and PLAEME
- NB: The PCMEP contains an unexpected Unicode character `ô` in `1350.DisMaryCross.psd`, line 1627. For now, this is circumvented by replacing the character with `o`, pending consultation with Richard Zimmermann for a different solution.

### 3.8 Version 8 (2019-01-31)

- Updated `lemmas.csv` and `frenchMEDverbs.csv`
- Added treatment of `~` (tilde) character

### 3.9 Version 7 (2018-10-08)

- Updated `lemmas.csv` to fix an error which caused lemmatised forms that were included in version 4 of the lemmatised corpus to be marked as NA. These forms are now marked as intended in version 8 of the lemmatised corpus.

### 3.10 Version 6 (2018-09-27)

- Changed the format of the form-lemma list from YAML to CSV
- This is more transparent and doesn't require users to install the `pyyaml` module in order to use the script
- Also, the table can be more easily edited and updated

### 3.11 Version 5 (2018-07-20)

- Updated version of `lemmas.yaml`

### 3.12 Version 4 (2018-07-18)

- Fixed a problem with the new file name format: round brackets “( )” replaced by square brackets “[ ]”
- Updated `lemmas.yaml` (fixed some wrongly lemmatised and unlemmatised forms)
- Updated list of French-based verbs
- Added a “Details” section in the present file, which provides information on the origin of the information inserted into the corpus

### 3.13 Version 3 (2018-04-16)

- Included an updated (partially disambiguated and manually corrected) list of form-lemma links
- Resulting file names are now in a new format that begins with the manuscript date

### 3.14 Version 2 (2018-02-05)

- Included most recent list of form-lemma links and French-based verbs
- Summary option is now activated via an argument call (`-s` or `--summary`) rather than uncommenting specific lines of code

### 3.15 Version 1 (2016-07-25)

- Version of `insertLemmas.py` made available for download on the BASICS Toolkit web application
  - Changes: the creation of a summary file by Middle English sub-period (M1-M4) and text base (en/fr) is deactivated, as this requires renaming the corpus files to include such information in the file name (e.g. by renaming “`cmaelr3.m23.psd`” as “`m2-en.cmaelr3.m23.psd`”)
  - Users wishing to reactivate the summary output after renaming the .psd files can do so by uncommenting the following lines in the script (by removing the initial “`#`”): lines 400-401, lines 403-405, lines 407-408, 410-420
  - See below for log of development version of the script
- 

## 4 Documentation for the development version of `insertLemmas.py`

### 4.1 Version 9 (2017-10-17)

#### 4.1.1 Changes from previous version:

- Updated list of French-based verbs with the following verbs:
  - `define`, v.
  - `de'liber`, v.
  - `disappoint`, v.
  - `disdain`, v.
  - `'herald`, v.
  - `intend`, v.
  - `travail`, v.
  - `joy`, v.
  - `solace`, v.

- Updated version of `lemmas.yaml`, which includes previously unlemmatised verbs that were found when analysing verbs prefixed with {a-}

## 4.2 Version 8 (2017-07-28)

### 4.2.1 Changes from previous version:

- Updated `lemmas.yaml`, fixing lemmatisation errors noticed during data annotation
- Updated list of French-based verbs, including gaps in the OED-MED links, and verbs not included in the OED list

## 4.3 Version 7 (2017-01-12)

### 4.3.1 Changes from previous version:

- Added `e=nonfrench` label

## 4.4 Version 6 (2016-07-01)

### 4.4.1 Changes from previous version:

- Updated version of `lemmas.yaml`

## 4.5 Version 5 (2016-06-06)

### 4.5.1 Changes from previous version:

- Changed lemma insertion format to have @ at beginning as well as at the end
- Changed format of summary output file so that it can be treated directly in R
- Added more spelling variation rules
- Updated version of `lemmas.yaml`

## 4.6 Version 4 (2016-04-02)

### 4.6.1 Changes from previous version:

- Changed script to handle V tags only
- Changed script to deal with split verbs, e.g. (VBP21 bi@l=NA@m=NA) (VBP22 tacne+d@l=toknen@m=46220) is now treated as (VBP21 bi) (VBP22 tacne+d@l=bitoknen@m=4955)
- Added handling of verb forms preceded by \$, e.g. \$dreden
- Added use of the stem of unrecognised forms to match stems of recognised forms

## 4.7 Version 3 (2016-02-09)

### 4.7.1 Changes from previous version:

- Switch verb form to lowercase before looking for matching lemma, to prevent cases of no-match for verb form in all-caps or at the beginning of sentences
- Removed handling of MD (modal verbs), as this yielded NAs only
- Some MED lemmas have brackets to mark variants, e.g. `assail(l)en`, which interfered with Penn-format, changed to square brackets: `assail[l]en`

## 4.8 Version 2 (2016-02-05)

### 4.8.1 Changes from previous version:

- Updated version of `lemmas.yaml`
- Implemented spelling variants for verb form with no matching lemma

## 4.9 Version 1 (2016-01-14)

### 4.9.1 What it does:

Inserts lemma information in verbs of PPCME2.

### 4.9.2 Prerequisites:

- Python Version 2.7.x
- `yaml` (Python module: can be installed with `pip install pyyaml`)
- Input files:
  - a directory containing the PPCME2 corpus, placed in the same directory as the script file
  - `lemmas.yaml`
  - `frenchMEDverbs.csv`

### 4.9.3 Usage:

- Open Terminal
- Change working directory to `InsertLemmas` directory:

```
cd ~/basicsdata/input/michaelpercillier/InsertLemmas
```

NB: Change path accordingly if the folder is copied elsewhere. Alternatively, you can drag the folder to the Terminal window after having typed “`cd`”

- Launch script:
  - either specify the name of the corpus folder in the program call:  
`python insertLemmas.py corpusDir`
  - if only `python insertLemmas.py` is called, the corpus directory name will be asked from the user

#### 4.9.4 Output:

- a directory, named after the input directory and the suffix `_Lemmatised`
  - inside this directory, the corpus files, named after the input files, with `.lemma.psd` ending
  - in these files, all verb forms contain the following information
    - \* lemma suggestion, marked by `@l`
    - \* MED number of the lemma suggestion, marked by `@m`
    - \* French etymology, marked by `@e=french` when applicable
    - \* `@w=doubt` when coders marked they had doubts about their lemma suggestion(s)
    - \* NB: verb forms can have more than one lemma suggestion
- a summary file, displaying for each file (and globally):
  - the total number of verbs
  - the number of verb forms with no matching lemma
  - the number of verbs with French etymology
  - the number of verbs where coders made multiple lemma suggestions or marked that they had doubts