Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Using CLDFBench and PyLexibank on Windows

David Snee
Chair for Multilingual Computational Linguistics
University of Passau

Using tools such as CLDFBench and PyLexibank, datasets can be converted into Cross-Linguistic Data Formats (CLDF), offering a standardized and interoperable representation of linguistic data. While these tools are powerful, lifting datasets to CLDF can present unique challenges for Windows users due to idiosyncrasies in the Windows operating system. Although CLDFBench and PyLexibank are compatible with Windows, certain workarounds may be necessary to address system-specific issues. This guide aims to demonstrate how CLDFBench and PyLexibank can be effectively installed and used on a Windows computer to lift a dataset to CLDF.

1 Introduction

Lexibank is a large repository of multilingual wordlists with unified transcriptions and semantic glosses available for over 2,000 languages (https://lexibank.clld.org, List et al. 2022). In order to unify the underlying wordlist data, Lexibank makes use of Cross-Linguistic Data Formats (CLDF), an attempt to offer consistent standards for the coding of various kinds of cross-linguistic data (https://cldf.clld.org, Forkel et al. 2018).

In order to convert a dataset to the CLDF standards, the CLDFBench package (https://pypi.org/project/cldfbench, Forkel and List 2020) can be used. CLDFBench is a Python package that can be invoked from within Python scripts and from the commandline. It facilitates the CLDF conversion by offering a clear workflow by which original datasets — which can be provided in all different kinds of formats — are converted systematically into unified CLDF packages. For lexical data, there is an additional plugin, called PyLexibank (https://pypi.org/project/pylexibank, Forkel et al. 2021), which adds additional functionalities for the creation of wordlists in CLDF, implementing additional standards needed for wordlists published as part of the Lexibank repository. Most importantly, PyLexibank extends the integration with reference catalogs by providing direct access not only to Glottolog (https://glottolog.org, Hammarstrom et al. 2024) for information on language varieties, but also to Concepticon

Computer-Assisted Language Comparison in Practice
Volume 7, Number 2
URL: https://calc.hypotheses.org/7852 DOI: 10.15475/calcip.2024.2.6
Published under a Creative Commons Attributions 4.0 LICENSE
Published on 18/12/2024

(https://concepticon.clld.org, List et al. 2024a) for the handling of semantic glosses, and CLTS (https://clts.clld.org, List et al. 2024b) for the handling of phonetic transcriptions.

The basic workflow for the conversion of multilingual wordlist data to standardized CLDF formats required by the Lexibank repository has been outlined in previous studies (List 2021, Bocklage et al. 2024, Miller and List 2024). However, implementing this workflow on Windows can present challenges due to certain peculiarities of Windows operation systems that tend to throw errors that are difficult to understand when following the traditional installation instructions oriented towards users with Unix operation systems. In the following, we will give a short overview on the most important aspects that need to be kept in mind when running CLDFBench and PyLexibank on Windows. The steps reported here have all been successfully tested on the operation system Windows 10.

2 Major Problems on Windows

From what we have seen so far, there are two major problems with the installation of CLDFBench and PyLexibank on Windows. On the one hand, one must make sure to explicitly allow for long file names on one's Windows system. On the other hand, one must avoid using standard configuration files offered by CLDFBench, since these are not well integrated on Windows.

Using PyLexibank and CLDFBench successfully requires not only to install Python along with the the respective Python packages, but also to download the reference catalogs that build the backbone of the standardization. The recommended tool to obtain these catalogs is the version control system GIT (https://git-scm.com/). GIT also runs on Windows, and it is not difficult to install it, but when trying to download data underlying Glottolog, Windows users usually run into the problem that the length of file names on Windows is usually restricted to 260 characters (which Glottolog data easily exceeds).

Additionally, Windows users may run into difficulties, resulting from the fact that configuration information for packages installed on a computer are handled differently in Windows compared to Unix systems. While CLDFBench has been programmed in such a way that such differences should not cause any particular problems, we found that some extra precautions are needed when trying to install and use CLDFBench and PyLexibank on a Windows computer.

Thus, while there are workarounds for all problems that may occur, Windows users will — unfortunately — have to do some extra work in order to get PyLexibank and CLDFBench running on their systems. In the following, we will list these in due detail.

3 Preparing Windows for CLDFBench and PyLexibank

3.1 Installing Preliminary Packages

In order to get started, we assume that you have Python (https://python.org, 3.8 or higher) and GIT installed on your system. We also assume that you have installed the Powershell (https://learn.microsoft.com/powershell/) that provides an enhanced commandline access in Windows and know how to start it, and — if needed — how to run it as administrator.

3.2 Handling Long File Names

One of the crucial steps before trying to download and install any packages is to ensure that you have long file paths enabled. This can be done in a quite straightforward manner by following the tutorial by Glenn and Lewis (2023). Since this tutorial will run interested users through all necessary steps, we won't discuss any of the steps here in detail, but refer interested users to the tutorial itself.

3.3 Installing CLDFBench and PyLexibank

If you have Python installed, you have also access to PIP (https://pypi.org/project/pip/), the package manager of Python that is typically used to install packages. We additionally and strongly advise all users to make use of a virtual environment when installing Python packages. We assume that you run your Python scripts in a folder called lexibank and that your username is dummy. If you open the terminal in that folder (or navigate to the folder using the cd command), the following lines will create a virtual environment called lexi and directly activate it, which means that any packages that you install will only work in those sessions in which you explicitly activate the respective environment.

```
PS C:\Users\dummy\Desktop\lexibank> python -m pip install
virtualenv
PS C:\Users\dummy\Desktop\lexibank> virtualenv lexi
PS C:\Users\dummy\Desktop\lexibank> Set-ExecutionPolicy -
ExecutionPolicy Unrestricted -force
PS C:\Users\dummy\Desktop\lexibank> .\lexi\Scripts\activate
```

Having set up the virtual environment in this way, you can now install the Python packages you need to convert lexical data to CLDF. You can achieve this through a single command on the commandline, as shown below. The prompt in the Powershell starts with (lexi) now, indicating that the virtual environment lexi was activated successfully.

```
(lexi) PS C:\Users\dummy\Desktop\lexibank> python -m pip install
pylexibank
```

This command will not only install the PyLexibank plugin, but with it also the CLDFBench package, along with a larger list of dependencies. After successful installation, you can test the installation by typing the following command into your Powershell terminal.

```
(lexi) PS C:\Users\dummy\Desktop\lexibank> cldfbench help
```

The output will show all subcommands currently installed along with CLDFBench, including many commands that start with `lexibank`, indicating that these are the commands that are only available after installing the PyLexibank plugin.

3.4 Cloning Reference Catalogs

There is a very convenient command in CLDFBench that allows you to download all three major reference catalogs (Glottolog, Concepticon, and CLTS) at once and store the information of where they reside on your computer in a configuration file that can later be invoked in your Python scripts. However, since the location of these configuration files also turned out to cause constant errors, we now recommend exclusively to store reference catalogs in an explicit location and to refer to them explicitly when invoking CLDFBench from the commandline. Let us — to get started — assume that you have created a directory called cats in the lexibank directory. The following lines will first enter this directory and then download all three packages with the help of GIT.

```
(lexi) PS C:\Users\dummy\Desktop\lexibank> cd cats
(lexi) PS C:\Users\dummy\Desktop\lexibank\cats> git clone
https://github.com/glottolog/glottolog.git
(lexi) PS C:\Users\dummy\Desktop\lexibank\cats> git clone
https://github.com/concepticon/concepticon-data.git
(lexi) PS C:\Users\dummy\Desktop\lexibank\cats> git clone
https://github.com/cldf-clts/clts.git
(lexi) PS C:\Users\dummy\Desktop\lexibank> cd ..
```

If you now open the folder cats, you should be able to see that there are three new directories inside the folder, called `glottolog`, `concepticon-data` and `clts`, respectively. If you wanted, you could change their names without destroying anything, but you need to remember where they are on your computer, in order to point to them when invoking CLDFBench from the commandline.

4 Example

The following example allows you to test if your Windows system has been prepared appropriately to allow you to use PyLexibank and CLDFBench without switching to alternative operating systems. We will use GIT to download an existing Lexibank repository and then rerun the command by which the original data is compiled to CLDF. We use the repository HattoriJaponic, based on Hattori's data on Japonic language varieties from 1973 (Hattori 1973), originally prepared as a test set for new methods for sequence comparison in historical linguistics by List (2014).

```
(lexi) PS C:\Users\dummy\Desktop\lexibank> git clone
https://github.com/sequencecomparison/hattorijaponic.git
```

While this dataset contains readily converted CLDF data already, it also contains the source code, in the form of a Python script that can be invoked by the CLDFBench package, that systematically converts the data from its raw form to CLDF. In order to do so, you must run the CLDFBench subcommand lexibank.makecldf in the terminal. When doing so, however, you must pass the information on where the three reference catalogs Glottolog, Concepticon, and CLTS can be found, and which versions of these should be used for the conversion. This information is passed in the form of specific arguments, traditionally preceded by two dashes `--` in commandline programs (take, for example, `--glottolog=PATH`). The full command that converts the raw data to CLDF looks as follows.

```
(lexi) PS C:\Users\dummy\Desktop\lexibank> cldfbench
lexibank.makecldf hattorijaponic\lexibank_hattorijaponic.py
--glottolog=cats\glottolog --concepticon=cats\concepticon-data
--clts=cats\clts --glottolog-version=v5.0
--concepticon-version=v3.2.0 --clts-version=v2.3.0
```

In the case of success, the output of this command should look similar to the following output (with personal paths removed here).

```
INFO running _cmd_makecldf on hattorijaponic ...
INFO file written: C:/Users/dummy/Desktop/lexi/hattorijaponic/cldf/.transcription-
report.json
INFO Summary for dataset C:\Users\dummy\Desktop\lexi\hattorijaponic\cldf\cldf-
metadata.json
- **Varieties:** 10 (linked to 10 different Glottocodes)
- **Concepts:** 200 (linked to 200 different Concepticon concept sets)
- **Lexemes:** 1,986
- **Sources:** 1
- **Synonymy:** 1.00
- **Cognacy:** 1,986 cognates in 460 cognate sets (182 singletons)
- **Cognate Diversity:** 0.15
- **Invalid lexemes:** 0
- **Tokens:** 9,042
- **Segments:** 65 (0 BIPA errors, 0 CLTS sound class errors, 65 CLTS modified)
- **Inventory size (avg):** 36.30
INFO file written:
C:/Users/dummy/Desktop/lexi/hattorijaponic/TRANSCRIPTION.md
INFO file written: C:/Users/dummy/Desktop/lexi/hattorijaponic/cldf/lingpy-
rcParams.json
INFO ... done hattorijaponic [940.9 secs]
```

5 Conclusion

It is by no means impossible to run workflows for computer-assisted language comparison on Windows. For complex packages, such as CLDFBench and PyLexibank, some additional preparations are needed.

References

Bocklage, K., Di Natale, A., Tjuka, A., and List, J.-M. 2024. "Representing the Database of Semantic Shifts by Zalizniak et al. from 2024 in Cross-Linguistic Data Formats." Computer-Assisted Language Comparison in Practice 7 (1): 25–35. DOI: 10.15475/calcip.2024.1.4.

Forkel, R.; Greenhill, S. J.; Bibiko, H.-J.; Rzymski, C.; Tresoldi, T. & List, J.-M. (2021): PyLexibank: The Python Curation Library for Lexibank [Software, Version 2.8.2]. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: https://pypi.org/project/pylexibank.

Forkel, R., List, J.-M., Greenhill, S. J., Rzymski, C., Bank, S., Cysouw, M., Hammarstrom, H., Haspelmath, M., Kaiping, G. A., and Gray, R. D. 2018. "Cross-Linguistic Data Formats, Advancing Data Sharing and Re-Use in Comparative Linguistics." Scientific Data 5 (1): 1–10. DOI: 10.1038/sdata.2018.205

Forkel, R., and List, J.-M. (2020): CLDFBench: Give Your Cross-Linguistic Data a Lift. In Proceedings of the Twelfth Language Resources and Evaluation Conference, 6995–7002. Marseille: ELRA. URL: https://aclanthology.org/2020.lrec-1.864.

Glenn, W., and Lewis, N. (2023): How to Make Windows 10 Accept File Paths over 260 Characters. How-To Geek, 22.08.2023. URL: https://www.howtogeek.com/266621/how-to-make-windows-10-accept-file-paths-over-260-characters/.

Hammarstrom, H., Forkel, R., Haspelmath, M., and Bank, S. (2024): Glottolog [Dataset, Version 5.0]. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: http://glottolog.org.

Hattori, S. (1973): Japanese Dialects. In H. M. Hoenigswald and R. H. Langacre Diachronic, Areal and Typological Linguistics, The Hague and Paris: Mouton. 368–400.

List, J.-M (2014): Sequence Comparison in Historical Linguistics. Dusseldorf: Dusseldorf University Press. URL: https://sequencecomparison.github.io.

List, J.-M. (2021): Converting the Vietic Dataset by Sidwell and Alwes from 2021 to CLDF. Computer-Assisted Language Comparison in Practice 4 (2). DOI: 10.58079/m6la.

List, J.-M., Forkel, R., Greenhill, S. J., Rzymski, C., Englisch, J., and Gray, R. D. (2022): Lexibank: A Public Repository of Standardized Wordlists with Computed Phonological and Lexical Features. Scientific Data 9 (1): 1–16. DOI: 10.1038/s41597-022-01432-0.

List, J.-M., Tjuka, A., van Zantwijk, M., Blum, F., Ugarte, C. B., Rzymski, C., Greenhill, S. J., and Forkel, R. (2024a): Concepticon [Dataset, Version 3.2.0]. Leipzig: Max Institute for Evolutionary Anthropology. URL: https://concepticon.clld.org.

List, J.-M., Anderson, C., Tresoldi, T., Rzymski, C., and Forkel, R. (2024b): Cross-Linguistic Transcription Systems [Dataset, Version 2.3.0]. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: https://clts.clld.org.

Miller, J. and List, J-M. (2024): Adding Standardized Transcriptions to Panoan and Tacanan Languages in the Intercontinental Dictionary Series. Computer-Assisted Language Comparison in Practice, 7.2: 69-77. DOI: 10.15475/calcip.2024.2.3.

Funding Information

This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101044282). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.