Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Adding data sets to NoRaRe: A guide for beginners

Annika Tjuka
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

There has been much discussion about the reproducibility of research and how it can be improved (Munafò et al. 2017). One antidote to the "reproducibility crisis" seems obvious: data sharing. However, one additional point that is not mentioned as often is the standardization of shared data to make them comparable. Especially for cross-linguistic studies, this is an important step that needs to take place so that we can conduct reproducible studies in different languages. The NoRaRe database (Tjuka et al. 2021a) is a resource that provides standardized cross-linguistic data on norms, ratings, and relations published in psychology and linguistics. In this blog post, I describe a beginner's guide to adding data sets to NoRaRe.

Introducing NoRaRe

The Database of Cross-Linguistic Norms, Ratings, and Relations for Words and Concepts (NoRaRe) combines data from psychology and linguistics. Since psychologists and linguists collect an increasing amount of data for a growing number of languages to describe various properties of words and concepts, we established this resource to make the available data comparable. The current version of NoRaRe (v0.2, Tjuka et al. 2021b) includes 65 unique word and concept properties derived from 98 different data sets across 40 languages.

The data is publicly available on GitHub and stored on Zenodo. To easily extend the database, we established three workflows to account for the different structures of the data. The workflows ensure that we can add data of different formats. The automated workflow intended for large-scale data sets of more than 2,000 items uses Python scripts and the command-line interface to automatically download and map a given word list (a description of all workflows can be found in Tjuka et al. 2021a). The following guide introduces a step-by-step instruction to follow the automated workflow.

Computer-Assisted Language Comparison in Practice
Volume 4, Number 8
https://calc.hypotheses.org/2690/

Published on 11/08/2021

Installation

If you haven't installed Python yet, this is the first thing you need to do. Before you install the NoRaRe data, it is advisable to set up a Virtual Environment. For more information on how to set up a virtual environment, you can check out my beginner's guide for adding concept lists to Concepticon. Since the data is stored in a GitHub repository, make sure that you have `git` installed on your computer. If you don't, you can use this tutorial to help you set it up. For the repository, you should create a new directory with the folder name `norare`, for example, by using the following command:

```
$ mkdir PATH/TO/concepticon
```

To download the repository, you need to visit https://github.com/concepticon/norare-data, click on the green button `code`, and copy the link. You can then clone the repository on the command line:

```
$ git clone https://github.com/concepticon/concepticon-data.git
```

In addition, you need to download the Concepticon (List et al. 2021, 2016) repository (https://github.com/concepticon/concepticon-data):

```
$ git clone https://github.com/concepticon/concepticon-data.git
```

Make sure that you either make a fork or create a branch so that you can open a pull request later on.

Apart from the data, you'll also need the Python package `pynorare` (List & Forkel 2020) to invoked commands directly from the command line. You can install the package by typing:

```
$ pip install pynorare
```

If everything is set up correctly, the following command should give you a list of all the commands and arguments of `pynorare`:

```
$ norare --help
```

When adding word lists to the NoRaRe repository, you will also need to have the Concepticon data stored. To define a default repository, you can open the configuration in a text editor, for example, `nano`:

```
$ nano /home/USERNAME/.config/cldf/catalog.ini
```

For Mac users it is

```
$ nano /Users/USERNAME/Library/Application\ Support/cldf/catalog.ini
```

In the opened document you can add clones:

```
[clones]
concepticon = /PATH/TO/concepticon/concepticon-data
```

Adding a new data set

As a data set, we define a word list in addition to its metadata, i.e. the list, the scripts for mapping the list, and the raw data. To add a new data set, you first need to create a new folder in `concept_set_meta`. The file name should consist of the first author's name, the publication year, and a key property, for instance, `Speed-2021-Sensorimotor`. The folder needs to contain a `map.py` file, an empty `raw` folder, and a `metadata.json` file with the name `Speed-2021-Sensorimotor.tsv-metadata.json`.

The `map.py` file includes the ID, a function to download the data, and a function to map the data to Concepticon. Depending on the structure of the data this file can be adapted. For example, part-of-speech tags can be added to improve the mapping, different file formats can be loaded with `get_excel` or `get_csv`, and so on. The `map.py` for the data by Speed and Brysbaert (2021) looks like this:

```
from pynorare.dataset import NormDataSet

class Dataset(NormDataSet):

  id = "Speed-2021-Sensorimotor"

  def download(self):
    self.download_file(
      'https://osf.io/wzfpd/download',
      'SpeedBrysbaert_Norms.xlsx'
    )

  def map(self, write_file=True):

    sheet = self.get_excel('SpeedBrysbaert_Norms.xlsx', 0, dicts=True)
```

```
    self.extract_data(
            sheet,
            gloss='DUTCH',
            language='nl',
            pos=True,
            pos_mapper = {
               'N': 'Person/Thing',
               'ADJ': 'Property',
               'WW': 'Action/Process',
               'Function': 'Other',
               'TW': 'Number'},
            pos_name = "DUTCH_POS"
            )
```

Next, you need to create a `metadata.json` file to define the content of the data set as well as the content of each individual column. Here, you can also decide which columns you would like to include in the final `.tsv` file that is mapped to the Concepticon concept sets. As an example, you can take a look at the `metadata.json` file from the data set `Speed-2021-Sensorimotor` here.

In addition, you need to fill in the information of your data set in the files `norare.tsv` and `conconcept_set_meta.tsv`. The former includes additional information on each column, for example, which rating scale was used to collect a given property. The latter includes a general description of the data set. The reference in BibTeX format is added to the `.bib` file in the folder `references/references.bib`.

If you have added all the necessary information, you can download and map the data via the command line by typing:

```
$ norare download Speed-2021-Sensorimotor
$ norare map Speed-2021-Sensorimotor
```

The second command creates a new `.tsv` file in the folder which includes the words that were mapped to a corresponding Concepticon concept set and the columns with the property values. To check if there are any inconsistencies and view the updated statistics of the NoRaRe data, you can use:

```
$ norare check
$ norare stats
```

If you have completed this guide until here without any errors, you are ready to add your data set to the NoRaRe repository by using `git push` and creating a pull request.

Outlook

The workflow described in this blog post is easy to handle and gives us the ability to add large data sets with only a few lines of code. By standardizing data offered across disciplines, we also contribute to the goal of enabling reproducible studies in multiple languages. The NoRaRe database will continue to grow in the future, and we have laid the groundwork for a collaborative enterprise between psychologists and linguists. I look forward to other researchers using the NoRaRe data and hope that this guide will facilitate the contribution to the database.

References

List, Johann Mattis & Rzymski, Christoph & Greenhill, Simon & Schweikhard, Nathanael & Pianykh, Kristina & Tjuka, Annika & Hundt, Carolin & Forkel, Robert. 2021. CLLD Concepticon 2.5.0. Geneva: Zenodo. https://doi.org/10.5281/zenodo.4911605.

List, Johann-Mattis & Robert Forkel. 2020. concepticon/pynorare: pynorare 0.2.0. Geneva: Zenodo. https://doi.org/10.5281/zenodo.3955051.

List, Johann-Mattis, Michael Cysouw & Robert Forkel. 2016. Concepticon: A resource for the linking of concept lists. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation, 2393–2400. Portorož, Slovenia: European Language Resources Association. https://aclanthology.org/L16-1379/

Munafò, Marcus R., Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware & John P. A. Ioannidis. 2017. A manifesto for reproducible science. Nature Human Behaviour 1(1). 1–9. https://doi.org/10.1038/s41562-016-0021.

Speed, Laura J. & Marc Brysbaert. 2021 (forthcoming). Dutch sensory modality norms. Behavior Research Methods. 1–38. psyarxiv.com/zv6pn.

Tjuka, Annika, Robert Forkel & Johann-Mattis List. 2021a. Linking norms, ratings, and relations of words and concepts across multiple language varieties. Behavior Research Methods. 1–21. https://doi.org/10.3758/s13428-021-01650-1

Tjuka, Annika, Robert Forkel & Johann-Mattis List. 2021b. NoRaRe. A database of cross-linguistic norms, ratings, and relations for words and concepts (Version 0.2). Jena: Max Planck Institute for the Science of Human History. https://doi.org/10.5281/zenodo.3957681.

Tjuka, Annika. 2020. Adding concept lists to Concepticon: A guide for beginners. Blog. Computer-Assisted Language Comparison in Practice. https://calc.hypotheses.org/2225 (28 December, 2020).