Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Volume 3

2020

Edited by Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max-Planck Institute for the Science of Human History
Jena

Online available at:
https://calc.hypotheses.org/posts-from-2020

Table of Contents

Computer-Assisted Language Comparison in Practice
Volume 3
https://calc.hypotheses.org/posts-from-2020

Published on 20/06/2021

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Automated Mapping of Metadata to Concepticon

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

While the core of the Concepticon project (https://concepticon.clld.org, List et al. 2019) are the numerous conceptlists which are constantly being added by the growing list of contributors, we have already from the beginning of the project, with the first version (List et al. 2016) tried to collect various kinds of concept metadata for all our concept sets.

Concept metadata is different from traditional concept lists, as metadata are potentially unlimited in size. Typical collections of age of acquisition data, for example, may easily count more than 5000 entries. Lists of this size are quite difficult to map to Concepticon, not only because it is tedious to manually correct and link thousands of entries, but also because concept metadata, as we find it in collections of norm data, is usually language-specific, which makes it even more difficult to find the best way to link a given word to a Concepticon concept set, since words can show a degree of ambiguity which we cannot find in the concept sets listed in the Concepticon so far.

Given these difficulties, we decided that we will allow for a less stricter treatment of links to concepticon when dealing with concept metadata. Assuming that most usecases involving metadata do not necessarily need the strict hand-curated Concepticon mappings which we provide for typical wordlists, we can therefore make use of automated approaches to identify the best matches for a given metadataset.

But how can we best identify these matches when dealing with a new dataset providing interesting concept metadata? While one could apply a simple brute-force procedure in which all concept sets in Concepticon are compared against a given concept metadataset, including fuzzy matchings and the like, I recommend a different approach which is extremely fast and at the same time accurate enough to provide the most obvious matchings of a given dataset against the data we have already linked to Concepticon.

Computer-Assisted Language Comparison in Practice

Volume 3, Number 1
https://calc.hypothses.org/2250

This procedure makes direct use of the multi-lingual Concepticon mappings which we automatically produce upon each new release of Concepticon. These mappings contain all elicitation glosses along with the concept sets to which they were linked for all glossing languages we have encountered so far in Concepticon (29 by now). The advantage of comparing a given collection of metadata directly with these mappings is that we can make active use of human judgments by which concepts were linked in the past.

In order to load these, we can make use of the new cldfcatalog API, which allows us to have convenient access to the Concepticon data on our system. We can install these along with the cldfbench package. Before we can start writing our Python script, we have to install cldfbench and configure our reference catalogs, by typing in the following two commands and following the instructions to which you will be prompted.

```
pip install cldfbench
cldfbench catconfig
```

Now we can start and load all mapping data from Concepticon in a fresh Python script. We first load the libraries we will need for this tasks:

```
from cldfcatalog import Config
from csvw.dsv import UnicodeDictReader
from collections import defaultdict
```

Now, we can load the Concepticon repository, which gives us the absolute path to the Concepticon data in our system.

```
repos = Config.from_file().get_clone('concepticon')
paths = {p.stem.split('-')[1]: p for p in repos.joinpath(
  'mappings').glob('map-*.tsv')}
```

We now create a mapping dictionary which will store all direct multi-lingual mappings that we can find in our Concepticon data. These are ordered by their priority, and we will do the same, as this will help us later to identify the best matches in those cases where there are more possibilities.

```
mappings = {}
for language, path in paths.items():
  mappings[language] = defaultdict(set)
with UnicodeDictReader(path, delimiter='\t') as reader:
  for line in reader:
```

```
    gloss = line['GLOSS'].split('///')[1]
    mappings[language][gloss].add(
      (line['ID'], int(line['PRIORITY'])))
```

For our mapping experiment, we use the data of Alonso et al. (2005) on age of acquisition in Spanish. The data are available for download from the publisher, but I had to modify them, since I encountered Unicode errors in the original version of the data. The resulting data file is called spanish-data.tsv and distributed along with the code accompanying this small tutorial.

To map the data, we have nothing else to do but to load the data and compare whether we find a direct match with the Spanish word in the original data and our list of mappings. The matches are storedin a specific dictionary (called esdata here).

```
esdata = defaultdict(list)
with UnicodeDictReader(
  'spanish-data.tsv',
  delimiter='\t') as reader:
  for i, line in enumerate(reader):
    if line['word'] in mappings['es']:
      best_match, priority = sorted(
        mappings['es'][line['word']],
        key=lambda x: x[1])[0]
      esdata[best_match] += [[
        str(i+1),
        line['word'],
        line['averageAoA'],
        best_match,
        priority]]
```

All we have to do now is to write the data to file. We need to make sure that links are unique, so we take only the best out of potential multiple matches.

```
with open('spanish-data-mapped.tsv', 'w') as f:
  f.write('ID\tWORD\tAOA\tCONCEPTICON_ID\tMATCH\n')
  for key, lines in esdata.items():
    best_line = sorted( lines, key=lambda x: x[-1])[0]
    best_line[-1] = str(best_line[-1])
    f.write('\t'.join(best_line)+'\n')
```

And for convenience, we can print out, how many matches we could actually find:

```
print('Found {0} direct matches in data.'.format(len(esdata)))
```

The resulting mappings can be found in the file spanish-data-mapped.tsv. All data and code, along with the dependencies are available from this public GitHub Gist.

References

Alonso, M. A. and Fernandez, A. and Diez, E. (2015): Subjective age-of-acquisition norms for 7,039 Spanish words. Behav Res Methods 47.1. 268-274.

Johann Mattis List and Christoph Rzymski and Simon Greenhill and Nathanael Schweikhard and Kristina Pianykh and Robert Forkel (2019): Concepticon. A resource for the linking of concept lists (Version 2.2.0). Max Planck Institute for the Science of Human History. Jena: https://concepticon.clld.org/.

List, Johann-Mattis and Cysouw, Michael and Forkel, Robert (2016): Concepticon. A resource for the linking of concept lists. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation. 2393-2400.

Cite this article as: Johann-Mattis List, "Automated Mapping of Metadata to Concepticon," in *Computer-Assisted Language Comparison in Practice*, 22/01/2020, https://calc.hypotheses.org/2250.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Adding concept lists to Concepticon: A guide for beginners

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

Scientific data should be openly accessible. This includes databases which are designed for collaborative work. However, in most cases, these databases are only extended by a team of experts. If a database is truly collaborative, the workflows need to be accessible for everybody. The Concepticon database (List et al., 2019) invites contributors to include their own data sets. This requires a transparent description of the contributing process.

Aim

Concepticon is a resource that stores concept lists. Those lists are hand-curated word lists which include concepts sets. The project is openly accessible with the intention to add new lists from different sources. But as of yet, the contribution process is rather tedious and often involves consultation of experts via the GitHub platform.

In this blog post, I present a step-by-step instruction for adding concept lists to Concepticon. In doing so, I hope that I can help those who wish to contribute to Concepticon to save some time. In addition, this post also provides a list of helpful links to other descriptions and further readings on the topic.

Installation

First, you should set up a Virtual Environment. Note that you need to install Python if it isn't already on your computer. You can create and activate the environment with the following commands:

```
$ virtualenv PATH/TO/NameOfYourEnvironment
$ source PATH/TO/NameOfYourEnvironment/bin/activate
```

Computer-Assisted Language Comparison in Practice

Volume 3, Number 1
https://calc.hypotheses.org/2225

The virtual environment is recommended because some of the packages, which you will install below, depend on each other. Thus, your commands will not work if one of them isn't installed. Another reason why you should use a virtual environment is that it installs all the packages only in the environment so that the Python library on your computer stays the same.

The basis for our Concepticon website is a GIT repository. Checkout this tutorial if you don't know how to set up GIT on your computer. For the repository, you should create a new directory with the folder name `concepticon`, for example, by using the following command:

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

You can then change into the folder and download or clone our repository here: `https://github.com/concepticon/concepticon-data`. With GIT, downloading the data is as simple as typing:

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

In addition, you need to install the `pyconcepticon` library:

```
$ pip install pyconcepticon
```

This will automatically install a terminal command concepticon on your computer, with which you can, among others, automatically map a concept list to Concepticon concept sets. Before you start, however, you need to make sure that the `pyconcepticon` toolkit knows where on your computer the `concepticon-data` folder is stored. While you can pass the location every time you use the command with help of the `--repos` argument, it is recommended to store this location in a configuration file on your computer. This file can be automatically installed with help of the `cldfcatalog` package, which is automatically distributed along with the `cldfbench` package:

```
$ pip install cldfbench
$ pip install pyglottolog
$ pip install pyclts
```

The last two commands install two major reference catalogs that are used as part of the Cross-Linguistic Data Formats initiative, namely Glottolog (Hammerstrom et al., 2019) and CLTS (List et al., 2019). Once you have done so, you can open the following file in the command line with a text editor of your choice and edit it. If you work in a Linux environment, for example, you can type:

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

Note that the configuration file's location may differ, with respect to your operation system. Please check the detailed instructions to the `appdirs` package, which allows to handle configuration files and application directories in a convenient manner across different platforms.

You can now manually add the default path that you want to use for the `pyconcepticon` package on your system, by inserting it into the file:

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

If this seems too complicated to be done upon first run, you can also actively pass the location of your `concepticon-data` folder when calling the `concepticon` commands such as `map_concepts`. In order to do so, just type the following and then the command you'd like to use:

```
$ concepticon --repos=PATH/TO/concepticon-data COMMAND
```

In the following, however, we will assume that you have configured the path via the catalog configuration procedure.

After the set up, you should have a folder with your virtual environment and the Concepticon repository, for example:

```
YOUR/USER/PATH/Projects/VirtualEnv/base-env/
YOUR/USER/PATH/Projects/Repos/concepticon/concepticon-data
```

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

```
$ concepticon --help
```

How can I map my list?

You can find an exhaustive instruction about the mapping process here: `https://calc.hypotheses.org/1820`. To prepare your list, follow the following steps:

- The file must have the following columns: GLOSS or ENGLISH, and NUMBER
- The list must be stored as a `tsv`-file, which is a tab-separated value text file ending in `.tsv`

- The name of the file should follow this scheme: `LastNameFirstAuthor-YearOfPublication-NumberOfWords`, e.g. `Mueller-2000-113.tsv`

The mapping of your list with the glosses in Concepticon will be done automatically when you type:

```
$ concepticon map_concepts PATH/TO/YOURLIST.tsv > test.tsv
```

If you decided not to use the configuration file that tells `pyconcepticon` where the Concepticon repository can be found, you can specify with `--repos=YOUR/PATH/TO/concepticon-data`, where you have stored your `concepticon-data` folder. If you open your terminal from within the folder `concepticon-data` (or you `cd`-ed into that folder), you do not need to specify the repository. The command `> test.tsv` saves your list as file `test.tsv` with three additional columns: CONCEPTICON_ID, CONCEPTICON_GLOSS, SIMILARITY. This file will appear in the folder where you opened your terminal.

Now, that you created an automated mapping, you can correct this by cleaning the list as follows: First, you should manually check each mapping proposed by the algorithm. If the algorithm only found one mapping, there will be a Concepticon ID and a Concepticon Gloss in the respective columns. If the algorithm did not find a mapping, this is indicated by three question marks `???`. If multiple matches were identified which are equally likely (based on the settings of the mapping algorithm), this is indicated by adding each possible mapping in one extra line, and adding one line which only has `#<<<` that indicates that a multiple mapping follows, as well as one line `#>>>` indicating that the multiple mapping has ended. You need to actively resolve all multiple mappings and should delete those lines which have wrong mappings. If none of the multiple mappings seems suitable, try to find a better mapping and add the respective ID and Gloss as provided by Concepticon (e.g., by searching on the Concepticon website, or in our specific app that allows a quick search in multiple languages), or you leave the Concepticon ID and Concepticon Gloss cells empty. In the end, there should be no field with question marks, no fields indicating the start or the end of a multiple mapping, and also no duplicate rows.

To finalize your mapping, remove the SIMILARITY column (which indicates the supposed quality of the match determined by the mapping algorithm), and create identifiers for you concept list with help of the link command (here applied for a fictitious list called `Mueller-2000-113.tsv`):

```
$ concepticon link Mueller-2000-113.tsv
```

Now, that mapping of the list has been finished successfully, you can place it into the `concepticon-data` folder and proceed to adding additional data and test the mapping.

Which files do I need to update in addition to my list?

If you add new concepts in your list, the following file needs to be updated: `PATH/TO/concepticon-data/concepticondata/concepticon.tsv`. Add the information of your list in the following file: `PATH/TO/concepticon-data/concepticondata/conceptlists.tsv`. If your list includes data from other authors, you can specify this in the NOTE column in the following format: `[Tischler 1999](:bib:Tischler1999)`. Update references for your list and secondary sources in: `PATH/TO/concepticon-data/concepticondata/references/reference.bib`. The BibTeX key (e.g., `Tischler1999`) for secondary sources must be equivalent to the one you inserted in the `conceptlists.tsv`.

Last but not least, add a `metadata.json` file for your list. You can create this file automatically with the following command:

```
$ concepticon create_metadata
```

This generates a `json`-file with the same name of your list in `PATH/TO/concepticon-data/concepticondata/conceptlists`.
Clean up your `metadata.json` file:

- If the file adds an additional GLOSS column which does not occur in your file, delete it in the `metadata.json`.
- Change the datatype of the columns according to your list. The command automatically adds the datatype `string` to some columns. But the data in your list columns might be of the following types:

i. `float` for floating-point numbers (e.g., 1.565765)
ii. `integer` for whole numbers (e.g., 1)
iii. `string` for characters (e.g., one)

- If a column has mixed values, for example, because the frequency values come from a secondary study, you may need to add the following:

```
{
"name": "FREQ_DLEX",
"null": "None",
"datatype": {
 "base": "float"
 }
}
```

Note that you can change the argument for “null” to any string which is used to indicate missing data: `NA`, `NaN`, etc.

How can I test if everything is well-integrated?

If you have done all of the steps above you can test your work with the following command:

```
$ concepticon test
```

The test spots potential errors. Examples would be:

i. There is only a Concepticon_ID or Concepticon_Gloss missing in a row.
ii. You inserted “Chinese” as a source language in `conceptlists.tsv` and your list does not have a column “CHINESE”.

Where can I find additional information?

Plenty of information on the mapping procedure has been published in the past. For a descriptions of the tags used in `conceptlists.tsv`, see List (2018). For additional information on how to contribute to the Concepticon project by making a pull request via GitHub, see the documentation supplied with the Concepticon GitHub repository.

References

List, Johann Mattis & Rzymski, Christoph & Greenhill, Simon & Schweikhard, Nathanael & Pianykh, Kristina & Forkel, Robert (eds.) 2019. Concepticon 2.2.0. Jena: Max Planck Institute for the Science of Human History. (Available online at http://concepticon.clld.org, Accessed on 2020-01-24.)

Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, Simon J. Greenhill, Christoph Rzymski, & Robert Forkel. (2019). Cross-Linguistic Transcription Systems (Version v1.2.0). Max Planck Institute for the Science of Human History. (Available online at http://glottolog.org, Accessed on 2020-01-24.)

List, J.-M. (2018): Towards a history of concept list compilation in historical linguistics. History and Philosophy of the Language Sciences 5.10. 1-14. URL: https://hiphilangsci.net/2018/10/31/concept-list-compilation/

Hammarstrom, Harald & Forkel, Robert & Haspelmath, Martin. 2019. Glottolog 4.1. Jena: Max Planck Institute for the Science of Human History. (Available online at http://glottolog.org, Accessed on 2020-01-24.)

Cite this article as: Annika Tjuka, "Adding concept lists to Concepticon: A guide for beginners," in Computer-Assisted Language Comparison in Practice, 29/01/2020, https://calc.hypotheses.org/2225.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Making an Annotated Concept List from the Data in CLICS

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

In this post I describe how the data from the CLICS project was used to make a new concept list for the Concepticon project.

The CLICS database in its current format makes direct use of the data assembled by the Concepticon project in order to aggregate lexical data from different sources. At the same time, the CLICS database itself can be seen as an interesting conceptlist, providing information on concept polysemy and semantic similarity.

For this reason, I added a CLICS conceptlist reflecting the results of the first version of the CLICS database (List et al. 2014) to one of the earlier versions of Concepticon. This concept list, called `List 2014 1280` offers different data on each of the Concepticon Concept Sets which were reflected in the first version of CLICS, as described in the following table.

Column	Description
Frequency	The number of languages in which a word for a given concept is attested.
Degree	The number of concepts with which another concept shares a colexification.
WeightedDegree	The sum of the number of different language families in which *any* colexification of the given concept is attested.
Rank	The rank, with respect to the degree of a given concept.
CommunityID	The identifier for the Infomap community that was computed for the data.
CommunityLabel	The *central concept* in the respective community, i.e., the concept with the highest degree.

Although CLICS has since then been updated two times (List et al. 2018, Rzymski et al. 2020), the corresponding concept lists have not yet been added for Concepticon. Since the creation of these concept lists involves some code, I decided to illustrate how one can produce the conceptlists from the CLICS datasets with some lines of Python code.

Computer-Assisted Language Comparison in Practice

Volume 3, Number 2
https://calc.hypotheses.org/2262

Code and Data Requirements

The code can be used for both the data underlying CLICS2 and the data underlying CLICS3. In both cases, the data is available in form of a file in GML-format, a rather flexible format for graphs, which contains the results of all calculations in form of annotations to the nodes and edges of the respective graphs. In both cases, the file that we will need is called infomap-3-families.gml. The file name reflects that the communities in the data were computed with help fo the Infomap algorithm (Rosvall and Bergstrom 2007) and that the threshold was set to 3 language families in order to accept a given colexification.

The data for CLICS2 can be found in the folder output/graphs/infomap-3-families.gml.zip at https://github.com/clics/clics2, and for CLICS3 it can be found in the main folder of the CLICS3 GitHub repository at https://github.com/clics/clics3 or in the CodeOcean capsule (https://codeocean.com/capsule/7201165/tree/v2, see under results/graphs) accompanying the study.

In the following, I will assume that you have downloaded the files and placed them in the same folder in which you also place the script to run the code. To make it easier to distinguish both files, I also assume that you rename them in clics2.gml and clics3.gml, respectively.

There are a couple of code requirements, such as LingPy (List et al. 2019), python-igraph (Csárdi and Nepusz 2006), and networkx (Hagberg 2009), which can be most easily installed with pip (see the installation instructions we provide for the CLICS3 package for details).

Before we start with the little script, we have to import the libraries we need.

```
import networkx as nx
import igraph
from lingpy.convert.graph import igraph2networkx
```

Note that we need igraph merely for loading the data in the GML format here, since the networkx package has a very strict requirement that all data be coded in ASCII. However, since this illustration uses networkx to iterate over the data and compute the missing data points, we will use lingpy‘s conversion function to convert the data to networkx in a second run.

We now add a little custom function that allows us to run the script with two different configurations, one for CLICS2, and one for CLICS3.

```
from sys import argv
if '2' in argv[1:]:
   clics = 'clics3'
```

```
    ref = 'List-2018'
else:
    clics = 'clics3'
    ref = 'Rzymski-2020'
```

This line will change the file we load (if we pass 2 as an argument, we will load clics2.gml, otherwise clics3.gml), and the name which we give to the identifiers in Concepticon and the file (since the authors differ for both lists).

Loading the graph

As mentioned before, we need to use the igraph package to load the graph. While we could also compute the relevant data points with this package alone, it is faster for myself to convert the graph to a networkx graph object, since I know the relevant functions better. In order to do so, we use the function as it is provided by LingPy. Given that LingPy expects a specific input format from the igraph graph, we need to give each node the attribute name, which is missing when loading the graph directly, but all in all this is a very small workaround.

```
_G = igraph.read(clics+'.gml')
for node in _G.vs:
    node['name'] = node['label']
G = igraph2networkx(_G)
```

Computing the degrees

One of the simplest way to compute polysemy scores from CLICS data is to compute the degree of each concept, i.e., the number of links it has to other concepts with respect to colexifications. In addition, we can also compute the weighted degree, which counts, how often a given colexification for a given concept occurs. Here, we have two possibilities: we can compute how many languages show a given colexification, or how many families.

```
deg = nx.degree(G)
fdeg = nx.degree(G, weight='FamilyWeight')
ldeg = nx.degree(G, weight='LanguageWeight')
```

Writing the data to file

We can now write the data to file. In order to do so, we create a simple list, in which we already insert the header.

```
table = [[
      'ID',
      'ENGLISH',
      'CONCEPTICON_ID',
      'CONCEPTICON_GLOSS',
      'FAMILY_FREQUENCY',
      'LANGUAGE_FREQUENCY',
      'WORD_FREQUENCY',
      'RANK',
      'COMMUNITY',
      'CENTRAL_CONCEPT',
      'DEGREE',
      'WEIGHTED_FAMILY_DEGREE',
      'WEIGHTED_LANGUAGE_DEGREE'
      ]]
```

While we have already computed the degrees, we will compute the rank on the fly, by sorting our graph according to the (unweighted) degree. The community refers to the identifier of the Infomap community which was computed for the respective study, and the central concept refers to the central concept in the respective community, measured by its (unweighted) degree. The information for both the community identifier and the central concept are both available in the graph we just loaded, as are the information on the frequency of the word (with respect to the attestation in the CLICS database, in form of words, languages, and as reflected in language families).

Filling the concept list with data is now straightforward. We just loop across the network, which we sort at the same time, and then add the relevant information to the table.

```
for i, (node, data) in enumerate(sorted(
      G.nodes(data=True),
      key=lambda x: deg[x[0]],
      reverse=True,
      )):
   table += [[
      '{0}-{1}-{2}'.format(ref, len(G), i+1),
      data['Gloss'],
      data['ConcepticonId'],
      data['Gloss'],
      str(int(data['FamilyFrequency'])),
      str(int(data['LanguageFrequency'])),
```

```
    str(int(data['WordFrequency'])),
    str(i+1),
    data['infomap'],
    data['CentralConcept'],
    str(int(deg[node])),
    str(int(fdeg[node])),
    str(int(ldeg[node])),
    ]]
```

Writing data to file

Once this is done, it is also only a three-liner to write the data to file (and it could be a oneliner, but I have no ambitions with respect to brevity here):

```
with open('{0}-{1}.tsv'.format(ref, len(G)), 'w') as f:
  for line in table:
    f.write('\t'.join(line)+'\n')
```

Submitting the data to Concepticon

In this form, little more information is required to make a pull request to the Concepticon at https://github.com/concepticon/concepticon-data and add the two new concept lists. The only things missing are

- the description of the concept list in conceptlists.tsv,
- the references for the two studies (they are available in BibTex from EvoBib),
- the metadata file in JSON format (which can be automatically created).

More information on how this can be done, can be found in the last months blog by Annika Tjuka (Tjuka 2020).

References

Gábor Csárdi and Tamás Nepusz (2006): The igraph software package for complex network research. InterJournal. Complex Systems .1695. .

Hagberg, Aric (2009): NetworkX. High productivity software for complex networks. http://networkx.lanl.gov/index.html.

List, Johann-Mattis and Greenhill, Simon and Tresoldi, Tiago and Forkel, Robert (2019): LingPy. A Python library for quantitative tasks in historical linguistics. Version 2.6.5. Max Planck Institute for the Science of Human History. Jena: http://lingpy.org.

List, Johann-Mattis and Prokić, Jelena (2014): A benchmark database of phonetic alignments in historical linguistics and dialectology. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation. 288-294.

List, Johann-Mattis and Greenhill, Simon J. and Anderson, Cormac and Mayer, Thomas and Tresoldi, Tiago and Forkel, Robert (2018): CLICS^2. An improved database of cross-linguistic colexifications assembling lexical data with help of cross-linguistic data formats. Linguistic Typology 22.2. 277-306.

Rosvall, M. and Bergstrom, C. T. (2008): Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences 105.4. 1118-1123.

Rzymski, Christoph and Tiago Tresoldi and Simon Greenhill and Mei-Shin Wu and Nathanael E. Schweikhard and Maria Koptjevskaja-Tamm and Volker Gast and Timotheus A. Bodt and Abbie Hantgan and Gereon A. Kaiping and Sophie Chang and Yunfan Lai and Natalia Morozova and Heini Arjava and Nataliia Hubler and Ezequiel Koile and Steve Pepper and Mariann Proos and Briana Van Epps and Ingrid Blanco and Carolin Hundt and Sergei Monakhov and Kristina Pianykh and Sallona Ramesh and Russell D. Gray and Robert Forkel and List, Johann-Mattis (2020): The Database of Cross-Linguistic Colexifications, reproducible analysis of cross- linguistic polysemies. Scientific Data 7.13. 1-12.

Tjuka, Annika (2020): Adding concept lists to Concepticon: A guide for beginners. Computer-Assisted Language Comparison in Practice 3.1. URL: https://calc.hypotheses.org/2225.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

RhyAnT: A Web-Based Tool for Interactive Rhyme Annotation

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

In times where home office is an obligation rather than an option, I have finally found time to create a first draft version of a web-based tool for interactive rhyme annotation. The tool is written in plain JavaScript, without any additional libraries, and supports the inline rhyme annotation format which we proposed in an earlier study. It allows for an efficient and save annotation of poems for their rhyme structure and will hopefully help us to assemble larger samples of rhyme patterns across genres, languages, times, and cultures.

Rhyme annotation is not the first thing a person will think of when asking a linguist what they do for a living. Nevertheless, poetry is a very interesting topic for linguistics, not only because it is realized in language, but also because it is so deeply influenced by communication, culture, and cognition. Additionally, poetry, or more specifically, rhyming, is of great importance for the reconstruction of Ancient Chinese pronunciation, since the Chinese characters alone would not provide us with enough hints to learn about their ancient pronunciations (Baxter 1992).

Starting from initial studies where I used rhyme networks to study the reconstruction of Old Chinese (List 2016, List et al. 2017), I have learned how important it is to be able to annotate what rhymes in a poem in an efficient way. My initial attempts, carried out also in close collaboration with Nathan W. Hill, consisted in the digitization of rhyme judgments on Ancient Chinese poetry collections (notably the Book of Odes), which I did in part by myself, in part with the help of student assistants. Back then, we realized that we would need an annotation format that would be both simple enough to be conveniently produced by various people while at the same time allowing for a very detailed annotation of the rhyme judgments that can be found in the literature.

Later in 2019, we found time to systematize these early attempts, proposing a first format that can be both annotated in inline fashion (meaning that you have an original text in front of you and add some information on top of it) and in stand-off fashion

Computer-Assisted Language Comparison in Practice

Volume 3, Number 3
https://calc.hypotheses.org/2380

(meaning that you place the annotation somewhere else). In the study in which we presented this format (List et al. 2019), I also presented a first Python library that could parse texts in both basic formats and allow for a quick handling of the data inside Python scripts. This library, called PoePy (List 2019) is available in a very initial draft version.

In order to test the two different formats, I started to create examples in which I showed how different poems can be annotated. I quickly realized that specifically the simple format for inline annotation was the most efficient way to annotate poems directly, even if it had some drawbacks, since it uses square brackets to indicate the rhyme group of a word.

Thus, if you want to annotate a little line, such as the refrain of Eminem's "Lose yourself", you would have to do it as follows:

```
@Artist: Eminem
@Title: Lose yourself

 You better lose yourself in the mu[a]sic,
 the moment You own_[it],
 you better never let it [b]go.
 You only get one [c]shot,
 do [c]not
 miss your chance to [b]blow
 This opportunity comes once in a lifetime.
```

The fact that there are two empty spaces preceding each line here indicates that we are dealing with the refrain of the poem. Lines that start without preceding spaces are treated as normal stanzas, and stanzas and refrain are separated by a blank line each. Additional metadata can be added in front of a poem by using a construct of @key:value. In this way, the German folk song Hanschen klein, could be annotated as follows:

```
@Author: Franz Wiedemann
@Source: Wikipedia
@Url: https://de.wikipedia.org/wiki/H%C3%A4nschen_klein
Hänschen [a]klein,
ging al[a]lein,
in die weite Welt hi[a]nein.

Stock und [b]Hut,
steh'n ihm [b]gut,
ist gar wohlge[b]mut.
```

```
Aber Mutter weinet [c]sehr,
hat ja nun kein Hänschen [c]mehr!

Da be[d]sinnt,
sich das [d]Kind,
läuft nach Haus ge[d]schwind.
```

For the quick annotation of poems, the stand-off format was never an option, I have to admit, since it shows such a high level of sophistication that it can only be used to store a given poem after the initial annotation has been done in this format, allowing scholars interested in details of rhyming to annotate additional aspects, providing alignments of rhymed sequences, and the like.

When I found out that one of my colleagues, Oleg Sobchuk, was not only personally fascinated by rap and hip-hop, but also scientifically, we decided to try and make a collection of annotated rap songs in German, Russian, and ideally also in English. During the first months, when we tested how well the inline annotation format could be used to consistently annotate poem after poem, I quickly realized how annoyed I always became when having to write another pair of square brackets and trying to remember which letter code I had been using for the rhyme pattern.

I was hoping to find a way to arrive at a more convenient way of annotating a poem without having to type too much. However, although I have some experience in writing interactive applications in JavaScript (List 2017), I did not know how to design the tool for rhyme annotation in such a way that it would really make the annotation process convenient. Last weekend, I finally had the flash of inspiration I was waiting for. By designing an interactive application in which one can annotate both the plain text in typing, while at the same time being able to modify the text interactively, one would have the possibility to test specific ideas for interactive annotation while at the same time never losing touch to the original data.

As a result, I managed to prepare a first prototype of a rhyme annotation tool, which I now call RhyAnT for the lack of a better name. The tool is distributed in form of a website at https://digling.org/calc/rhyant (but you can also download the code from GitHub and use it offline) and first offers a text field in which one can paste the poem one wants to annotate. Once having pasted the text (adding metadata, refrain annotation, and stanza separation manually), one can press the ESCAPE key at one's computer, and annotate the data in an interactive panel, as shown in the following figure.

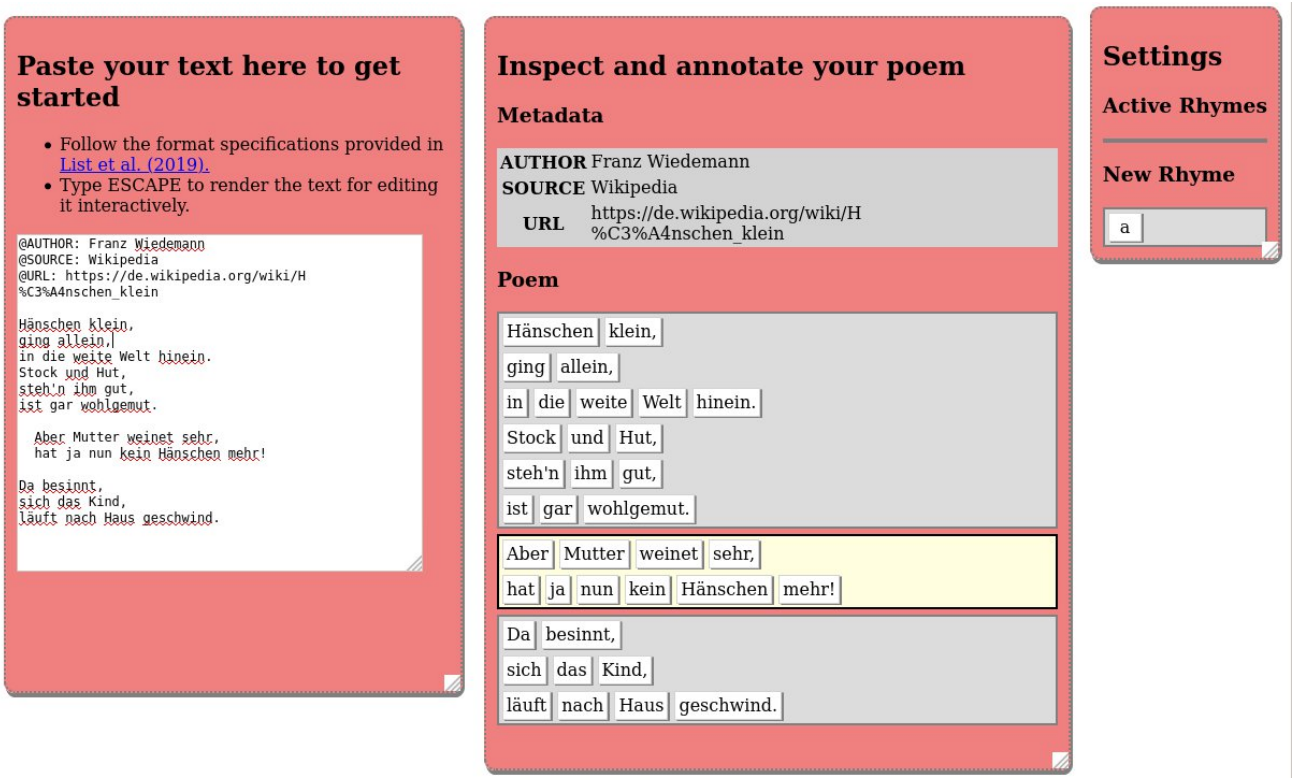

Figure 1: Loading an unannotated poem in RhyAnT

The annotation itself is pretty straightforward. To assign a word to a rhyme group, one has to select a “New Rhyme” from the top-right Settings panel first. Once this has been done, one can click on whatever word in the interactive annotation panel in order to assign it to that rhyme group. In order to switch to another rhyme, one just needs to click on the next “New Rhyme” again, or one can select one of the rhymes that have already been used. In order to de-select a rhyme assigned to a word, one just has to click a second time on it.

While annotating a poem interactively, the text in the original text panel will constantly be updated with the new annotations. At the same time, one can always modify the rhyme text directly, press ESCAPE again, and the interactive display will be updated. This allows not only to make sure that rhymes are properly added to syllables (by placing them before the syllable where they occur, such as I did when writinghin[a]nein in the Hanschen klein song above, which is not yet available in interactive mode), it also normalizes the rhyme text itself while annotating, making sure that the text can really be parsed by the tool. This nicely illustrates the core idea of interfaces in computer-assisted approaches, as it facilitates data annotation while securing that the annotation is correct at the same time.

While writing this text, I just used the tool to annotate the originally unannotated children’s song “Hanschen klein” with the tool. The following screenshot shows how this looks in action.

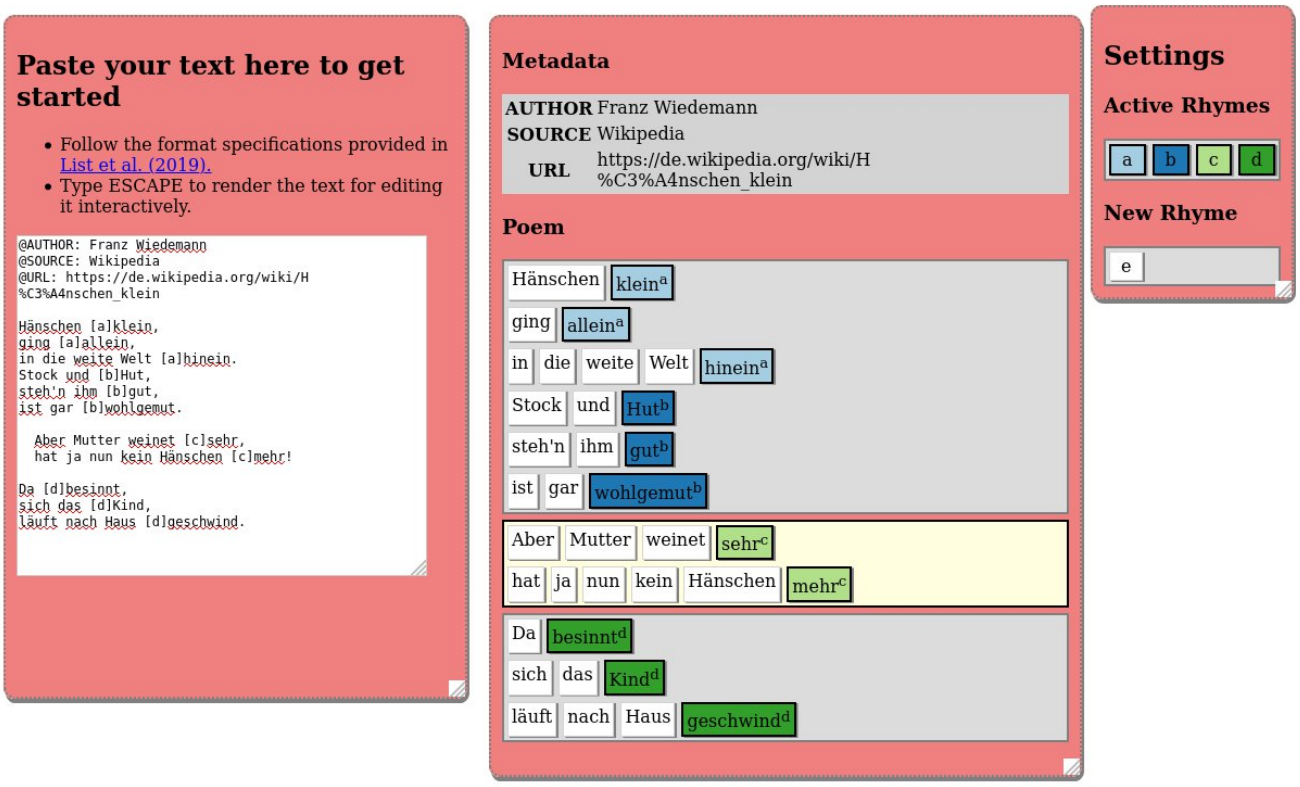

Figure 2: A readily annotated poem in RhyAnT.

The tool is not yet perfect and there are both some minor quirks that I would like to address in the future, as well as some bigger challenges that would allow for a more detailed annotation of rhymes. Among these, I would like to add the possibility to inspect all words that have been assigned to the same rhyme pattern. I would also like to have the possibility to add rudimentary alignments that would allow to be more specific on the parts of which one thinks that they make up for the rhyme pattern in question. Last not least, I would like to integrate this into some kind of a database system that would allow us to store annotated poems immediately and display them interactively on a website.

Whether this will be done any time soon is hard to tell, as I rarely find the leisure to code a whole Sunday. But I hope that even in this form the tool will help us to get closer to the dream of having a database of poetry across genres, languages, times, and cultures.

References

Baxter, William H. (1992): A handbook of Old Chinese phonology. Berlin:de Gruyter.

List, Johann-Mattis (2017): A web-based interactive tool for creating, inspecting, editing, and publishing etymological datasets. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. System Demonstrations. 9-12.

List, Johann-Mattis and Pathmanathan, Jananan Sylvestre and Hill, Nathan W. and Bapteste, Eric and Lopez, Philippe (2017): Vowel purity and rhyme evidence in Old Chinese reconstruction. Lingua Sinica 3.1. 1-17.

List, Johann-Mattis (2016): Using network models to analyze Old Chinese rhyme data. Bulletin of Chinese Linguistics 9.2. 218-241.

List, Johann-Mattis and Nathan W. Hill and Christopher J. Foster (2019): Towards a standardized annotation of rhyme judgments in Chinese historical phonology (and beyond). Journal of Language Relationship 17.1. 26-43.

List, Johann-Mattis (2019): PoePy. A Python library for handling annotated rhymes. Jena:Max Planck Institute for the Science of Human History.

Cite this article as: Johann-Mattis List, "RhyAnT: A web-based tool for interactive rhyme annotation," in Computer-Assisted Language Comparison in Practice, 25/03/2020, https://calc.hypotheses.org/2380.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

New Kusunda Data: A List of 250 Concepts

Uday Raj Aaley and Timotheus A. Bodt
[1]Independent Researcher, [2]SOAS University of London

Between 29th July 2019 and 12th August 2019, we invited the then remaining two speakers of the Kusunda language to Kathmandu, where we interviewed them. One of these speakers, Gyani Maiya Sen Kusunda, unfortunately passed away early 2020. At the moment of writing this, there is only one Kusunda speaker left, Kamala Khatri (Sen Kusunda).

We made a total of around 20 hours of video- and audio recordings. Part of our research involved the triple-repeated recording of a 250-concept word list from both speakers. The concepts themselves were taken from the original concept lists underlying the study of Sagart et al. (2019) on Sino-Tibetan languages. This means that comparisons of the new data for Kusunda can be conveniently done with any of the 50 languages in the sample reported by Sagart et al.

We make both this list and the original sound recordings available along with this contribution, and invite everyone to work with these data and communicate their findings. Any comments on the phonology of Kusunda are welcome, so that these can be used to further refine both the Roman and Devanāgarī orthographies used for writing and teaching Kusunda to the next generation of Kusunda speakers.

Out of the 250 concepts, the speakers did not remember 9 concepts and thought that 20 concepts did not exist. They had descriptive compounds for 3 concepts but did not agree on the exact form and they did not have consensus over the form in 7 other cases. A total of 10 concepts are confirmed Nepali loans (including all numerals 5-10). This left 200 concepts for comparison.

The 250-concept list consists of the 250 concepts with an ID in the first column, an English gloss in the second column, the identifier of Huáng's Tibeto-Burman Lexicon (1992), a link to the Concepticon project (List et al. 2020), a preliminary Kusunda reconstruction based on the available forms, comments on this reconstruction, a phonetic transcription of the form attested from Gyani Maiya with comments, and a phonetic transcription of the form attested from Kamala with comments.

The following table shows a small excerpt of the spreadsheet (note that columns and rows have been transposed here):

Computer-Assisted Language Comparison in Practice

Volume 3, Number 4
https://calc.hypotheses.org/2414

ID	33	34	35	36	37
ENGLISH	the dew	to die	to dig	dirty	the dog
Kusunda	∅	oɢ.da	mek	hu.wi.gɐn	ɐ.gəj
Comments1	does not exist, new compounds are made based on FALL + WATER, STAY + WATER etc.			< hu.wiː aː.gɐn, hu.wiː = ‘dirt, dust’, also verb ‘be dirty’	
Gyani Maiya	∅	ɔk.ɖaː	mʲɛk.tɔː	huj.gɐn	ɐ.gəj
Comments2	tɐŋ ʤɐk.ʤi water + stay i.e. ‘water which has accumulated on / in leaves’	ɔk.daː = die, ɔk.daː aː.gɔː = kill	mʲɛχ.tɔː, mʲɛʔ aː.gɔː		~ aː.gəj
Kamala	ʤʲun ʈɐŋ	ɔ̰.ɖaː	mʲɛk ɐ.gɔː	huj.gɐn	ɐ.gəj
Comments3	KGG_310719_A1; lit. FALL + WATER	‘to die (imp)’, ‘to kill’ ɔ̰ˤ.daː ɐ.gɔː (imp)	~ mʲɛːʔ.ɐɔː	< hu.wiː ɐ.gɐn, ~ huj.ʤiː; hu.wiː = dirt (n), dust	~ aː.gəj

The Kusunda reconstruction is based on the following observations and analyses:

- In general, word-initial affricates are palatalised in K, but non-palatalised in GM, whereas word-medial affricates are always non-palatalised in GM and sometimes non-palatalised in K. When preceding rhymes with vowel /i/, even GM *may* palatalise the word-initial and -medial affricates. Because all previous sources list only a dental affricate series [ʦ, ʦʰ, ʣ, ʣʰ], all affricates are thought to derive from this dental series, even when both speakers realise a palatal affricate [ʧ, ʧʰ, ʤ, ʤʰ].
- GM initial uvular stops [q, qʰ] corresponds to K initial uvular stops [q, qʰ]. Uvular stop [ɢ] has been completely lost word-initially in both speakers.

- Where a GM final [q] or [χ] corresponds to a K final [q] or [χ], this is thought to derive from underlying final *q.
- Where a GM final [χ], marginally [k] (sometimes preceded by a creaky vowel [y̰]) corresponds to a K creaky vowel [y̰ː] this is thought to derive from underlying final *ɢ.
- Where a GM intervocalic [χ] corresponds to a K intervocalic [q^h], this is thought to derive from underlying initial *q^h.
- Where a GM intervocalic [k^h] corresponds to a K sequence of creaky vowels [y̰.y̰], this is thought to derive from underlying initial *ɢ.
- Where a GM final [ŋ] corresponds to a K open nasalised vowel [ṽː] this is thought to derive from underlying final *ɴ.
- Where there is variation between vowel /o/, realised as [ɔ], and vowel /u/ in open syllables, this is thought to derive from an underlying vowel *u, with varying realisations depending on speaker. In closed syllables, variation between a [ɔ] and an [u] is thought to derive from an underlying vowel *o.
- Where a vowel /i/ varies with vowel /e/, the choice has been made to represent this in the ground form as *e, because vowel *i is preserved in both speakers.
- Where a rhyme [ɛ, ɛC_f] is preceded by a palatalised onset this is thought to derive from underlying form *e, *eC_f, respectively.
- Vowels /ɐ, ə/ are always short and hence vowel length has not been indicated in column 3.
- Length of vowels /i, ɔ, u/ is predictable on phonotactic position (long in open syllables, short in closed syllables) and has hence not been indicated in column 3.
- Vowel /a/ is always long and hence vowel length has not been indicated in column 3.
- Labialised onsets (e.g. sw-, gw-) are thought to be old and are hence indicated in the ground form.
- The off-glide in rhyme -ej is thought to be epenthetic and derive from an open rhyme *-e.
- Syllable-initially, both K's and GM's speech may show a simplification of diphthongs /əj/ and /ɐj/ to /ə/ and /ɐ/. Syllable-finally, both K's and GM's speech may show a simplification of diphthongs /əj/ to /a/. Where there is variation between K's [-əj] and GM's [-ɐj], -ɐj has been taken as underlying because K's [-əj] generally corresponds to GM's [-əj].
- Where GM's rhyme [-ɔw] varies in realisation with [-ɔː] and corresponds to K's rhyme [-uː], this is thought to derive from rhyme *-ow.

The data has been submitted to Zenodo, where it can be accessed in its version 1.0 via its DOI 10.5281/zenodo.3377537. We will be very happy for any kind of comments or suggestions, for which contact details can be found in the data we archived with Zenodo.

References

Huáng, Bùfán 黃布凡 (1992): {Z}àngmiǎn yǔzú yǔyán cíhuì 藏緬語族語言詞匯 [A Tibeto-Burman lexicon]. Běijīng 北京: Zhōngyāng Mínzú Dàxué 中央民族大学 [Central Institute of Minorities].(Available online at https://stedt.berkeley.edu/ stedt-cgi/rootcanal.pl/source/TBL)

List, Johann Mattis & Rzymski, Christoph & Greenhill, Simon & Schweikhard, Nathanael & Pianykh, Kristina & Tjuka, Annika & Wu, Mei-Shin & Forkel, Robert (eds.) 2020. Concepticon 2.3.0. Jena: Max Planck Institute for the Science of Human History. (Available online at http://concepticon.clld.org)

Sagart, Laurent and Jacques, Guillaume and Lai, Yunfan and Ryder, Robin and Thouzeau, Valentin and Greenhill, Simon J. and List, Johann-Mattis (2019): Dated language phylogenies shed light on the ancestry of Sino-Tibetan. *Proceedings of the National Academy of Science of the United States of America* 116. 10317-10322. DOI: 10.1073/pnas.1817972116.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

New Lexical Data for the Kusunda Language

Mei-Shin Wu
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

Endangered language documentation and endangered language revitalisation have been two hot topics in recent years. For instance, the United Nations Educational, Scientific and Cultural Organization (UNESCO) declared the year 2019 as the International Year of Indigenous Languages. However, although the UNESCO and many other organizations (e.g. The Endangered Languages Documentation Programme or SIL International) urge the public to be aware of the rapidly decreasing number of languages in the world, it does not slow down the annual rate of language loss. For example, the total number of speakers of the Kusunda language, a moribund language spoken in Nepal, decreased to only one person in 2020.

Introduction

I started collecting articles and lexical materials on the Kusunda language in 2017 with the hope to find time studying the origin of its speakers and its prehistoric contacts with neighboring languages. With the help of fellow scholars, I accumulated a total of 27 theses and articles. The earliest reference dates back to 1848 (Hodgson 1848). Unfortunately, I found only a handful of studies providing data on 100 or more lexical items. Subsequently, I converted Kusunda lexical material from five sources (Reinhard 1970; Rana 2002; Watters 2005; Donohue 2013; Aaley, 2017) into the formats recommended by the Cross-Linguistic Data Formats (CLDF, Forkel et al. 2018) initiative. In doing so, I met four major difficulties:

1. Data storing and sharing: Despite the fact that the idea of “open science” has been advocated for over a decade, “open data” is still an exception rather than the norm in the field of linguistics. Therefore, secondary linguistic data are either rare or fragmentary. Furthermore, “open” linguistic data are not persistently stored online.
2. Digitisation:
   Linguistic data accumulated over years but the majority of linguistic data is not available in digital form. This raises the problem that linguists cannot inspect the

Computer-Assisted Language Comparison in Practice

Volume 3, Number 4
https://calc.hypotheses.org/2246/

published data efficiently, and it also prevents linguists from applying further analysis on the data.

3. Standardisation:
   In Table 1, I provide three nouns and two verbs that frequently occur in linguistic studies: ‘cloud’, ‘hand’, ‘fog’, ‘to speak (1st SG)’ and ‘to move (1st SG)’. By just presenting five lexical items, one can already see wide variation in the existing Kusunda materials.
4. The phonological rendering of the words is not standardised according to International Phonetic Symbols (IPA). First, IPA has existed for more than 100 years, however, it is not being used regularly in existing historical linguistic data sets. Most of the time, we have encountered data recorded by customised phonetic symbols and accompanied by a guideline. For example, as Reinhard mentioned in his article, “j” and “ny” correspond to IPA /dz/ and /nʲ/. However, it is not always the case that a guideline is provided.
5. The basic vocabulary entries (Swadesh 1952), like ‘hand’ and ‘cloud’, have entirely different forms across the studies. Even though in the five studies conducted between 2002 and 2019 the lexical datasets were elicited from the same two speakers, Ms. Gyani Maiya and Ms. Kamala, the lexical material is highly diverse across the wordlists. Are these words the same words but in different “word forms” or do they represent the synonyms for the same concepts?
6. It commonly happens that linguists provide a list of words with English annotation but overlook the descriptions. It increases the difficulty in preparing large data sets when multiple datasets with divergent descriptions are involved. For example, the word for ‘hand; in Table 1 has various forms. I realised that /awəi/ means ‘hand’, ‘wrist’ and ‘arm’ after checking all the sources carefully. Another example is that English words like drink should mean ‘the drink’ or the ‘to drink’ Although linguists should be aware of the ambiguity of these glosses, they still use these glosses without giving any further explanations.
7. Attribution:
   There have been several instances, where data that were made available in open access in the public domain have been used and/or reproduced without proper attribution of the source and without proper credits to the original collector of the data. This may be a deterrent to other linguists to make their data available openly.

In the light of the outlined problems, rendering the existing datasets comparable becomes a task that consumes a lot of time and energy, with a low chance of success, since the task cannot be done without thinking of novel data representations that still keep trace of the original sources.

Gloss	Reinhard 1970	Rana 2002	Watters 2005	Donohue 2013*	Aaley 2017
cloud	duliŋ		bəm ~ pã:yi / pãi	pãj /pãj/	dʊliŋ
hand	tabi	nabi / amokh	awi / awəi	əwi /wi/	ɒmɒk, nabi, awi
fog	ganigiliŋ		dhundi		panji
to speak✢				məso /mso/	gipən ədʊ
to move (1st)✢	gauntsən		ghu a-t-ṇ, gho ə-go		ghʊ əgɒ

Table 1: The examples of existing data.* The format of Donohue's data is in "broad" and /phonemic transcription/. ✢ The glosses 'to speak' and 'to move' were not listed in the 200 items of the basic vocabulary list by Swadesh (1952).

New, comparable lexical data for Kusunda

Given the aforementioned problems, it was very nice to see the new data which Bodt And Aaley published on Kusunda last year. They interviewed the (maybe) last two speakers Gyani Maiya and Kamala and published their lexical data freely online. The project was sponsored by the Endangered Language Fund (ELF), the CALC research grant, as well as a crowdfunding enterprise. The original recordings and a short paper describing the process and the data are published on Zenodo (10.5281/zenodo.3377537), where they can be freely downloaded.

Additionally, a list of 250 basic vocabulary items (following the concept selection by Sagart et al. 2019) was prepared, archived with Zenodo, and presented in a short blog post (Aaley and Bodt 2020). In order to further enhance the comparability of the new Kusunda wordlist as published by Aaley and Bodt, we have now converted the original wordlist into CLDF format. The data in CLDF format themselves are curated on GitHub (https://github.com/lexibank/aaleykusunda, Version 1.0), and have also been archived with Zenodo (https://zenodo.org/record/3746946).

With the new data being available both in human- and machine-readable format, there is some hope that this fieldwork could inspire colleagues to start investigating the Kusunda language more closely or to help Aaley with his attempts to revitalize and document Kusunda. Additionally, the work addresses the four issues identified before:

1. Data storing and sharing:
   A short summary that describes the fieldwork method, along with the video and audio clips of the recordings are stored on Zenodo (https://zenodo.org/). Zenodo is a general-purpose open-access online archive, and it is widely used by scholars

to deposit their dataset and articles. All data and articles submitted to this website are given Digital Object Identifiers (DOIs), and as long as the dataset remains online, the given DOI will always point to the corresponding datasets (and Zenodo has been made for long-time archiving, so we do not talk about the next five years here only). Therefore, the new Kusunda data are always traceable and will be very hard to erase from the internet.

2. Digitisation:
   Aaley and Bodt did not have time to analyse all data they shared, instead they shared them openly in the hope to trigger the interest of colleagues who could help in analysing the data further. The transcribed Kusunda vocabulary is stored in CLDF format (https://cldf.clld.org, Forkel et al. 2018) in a public GitHub repository (https://github.com/lexibank/aaleykusunda), so the lexical material can be accessed without restriction. Since GitHub repositories are always in flux, and data may change, and it is not guaranteed that a proprietary provider, such as GitHub, will guarantee long-term-archiving, distinct versions of the data are archived (again) with Zenodo. Data in CLDF format is provided in plain text form in form of comma-separated values (CSV) and can be viewed not only in text editors, but also with common spreadsheet software, such as Excel or Google Sheets. It can also be curated and analysed by several Python libraries (notably, LingPy, https://lingpy.org, List et al. 2019), and many of these libraries provide detailed instructions with many usage examples (see e.g. List et al. 2018).
3. Standardisation:
   The primary goal of CLDF is to standardise linguistic data for the purpose of cross-linguistic comparison. In order to give a better view of the CLDF format, figure 1 outlines a simplified CLDF structure along with the roles distributed between linguists and computer programs. Table 2 and Table 3 are brief examples drawn from the data.
4. As shown in Table 2, each gloss is given a unique Concepticon gloss and a identification number (https://concepticon.clld.org, List et al. 2020). The clear definition can be found on the website. This strategy which regulates the usage of concepts can not only keep the data sheet “tidy”, but also gives a clear definition of the glosses.
5. As shown in Table 3, the lexical items are transcribed in IPA. There are three columns to hold different versions of transcriptions. Usually, the first column (Value) preserves the lexical items in the raw data, the second column (Form) holds the sound sequences as they are given in slight automatically preprocessed form, and the last column displays the results after conversion and tokenization. In this way, the raw form, if not written in IPA, can be preserved and the last

column gives the data for further analysis in software packages such as LingPy or annotation tools such as EDICTOR (https://digling.org/edictor/, List 2017).

6. Attribution:
   Like many academic articles often include a contribution section to detail the roles of each author, the repository of this dataset on Github also gives a CONTRIBUTORS.md file in which details the author (the field linguists), and the curator (also known as the maintainer). This keeps transparency of the source, and it shows appreciation to people who contribute their time and efforts in curating the data as well as keeping the data updated. Also, the new Kusunda dataset provides extensive instructions on how to cite the data in both bibtex format (the sources.csv in Figure 1) and the plain text form.

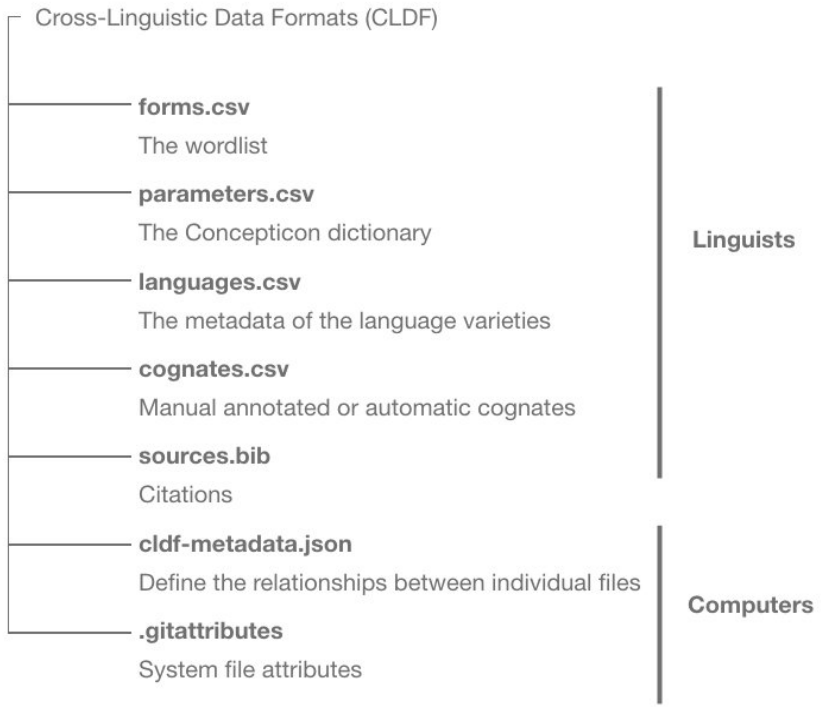

Figure 1: A brief description of the Cross-Linguistic Data Format structure.

ID	Name	Concepticon_ID	Concepticon_Gloss	Definition (Concepticon)
25	the Cloud	1489	CLOUD	https://concepticon.clld.org/parameters/1489
74	to give	1447	GIVE	https://concepticon.clld.org/parameters/1447
81	the hand	1277	HAND	https://concepticon.clld.org/parameters/1277

Table 2: An example of the Concepticon dictionary (parameters.csv). The format includes parameters IDs (ID), the gloss (Name), the concepticon IDs (Concepticon_ID), and the concepticon concepts

(Concepticon_Gloss). The links at the last column are not included in the data, as these links only provide convenience to retrieve the definition on the Concepticon website.

ID	Language_ID	Parameter_ID	Value	Form	Segments
KusundaGM-25-1	KusundaGM	25	bɐm	bɐm	b ɐ m
KusundaK-25-1	KusundaK	25	bɐm	bɐm	b ɐ m
Kusunda-25-1	Kusunda	25	bɐm	bɐm	b ɐ m
KusundaGM-74-1	KusundaGM	74	eː.gɔː	eː.gɔː	eː + g ɔː
KusundaK-74-1	KusundaK	74	eː.guː	eː.guː	eː + g uː
Kusunda-74-1	Kusunda	74	e	e	e
KusundaGM-81-1	KusundaGM	81	aː.wəj	aː.wəj	aː + w ə j
KusundaK-81-1	KusundaK	81	aː.wəj	aː.wəj	aː + w ə j
Kusunda-81-1	Kusunda	81	a.wəj	a.wəj	a + w ə j

Table 3: An example of the wordlist (forms.csv). The format includes unique entry IDs (ID), speaker or language IDs (Language_ID), the gloss ID (Parameter_ID, see the ID column in Table (2), the phonological rendering (Value), the phonetic sequence before tokenization (Form), and the tokenized phonetic sequences (Segments). The informants are Gyani Maiya (KusundaGM), Kamala (KusundaK), and the reconstructed proto-Kusunda words (Kusunda).

While people are complaining that linguistic data are not “open” enough, the way the new data are presented is a good example that shows that linguistic data can indeed be provided in transparent form. In addition the fieldwork done by Aaley and Bodt helps preserve the Kusunda culture, which is an important factor for a group’s ethnic identity. More work has to be done, and ideally, all recordings would be analysed and glossed, but it is obvious that this cannot be done immediately, but will require more time. Finally, the innovative ways used by Aaley and Bodt to obtain funding for their fieldwork via a crowdfunding campaign might also help to attract attention from a wider audience and encourage more scholars to work on language preservation.

I am very glad to see the new Kusunda data being presented openly on the internet, and I look forward to seeing more linguists to further work with the data, and come to new analyses as well as conclusions.

Acknowledgments

I thank Timotheus A. Bodt , Yunfan Lai, Sandra Auderset, Ilia Chechuro and Johann-Mattis List for providing information and comments.

References

Aaley, Uday Raj. 2017. Kusunda Tribe and Dictionary. Uday Raj Aaley.

Uday Raj Aaley and Timotheus A. Bodt. 2020. “New Kusunda data: A list of 250 concepts,” in Computer-Assisted Language Comparison in Practice, 08/04/2020, https://calc.hypotheses.org/2414.

Forkel, Robert, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarstrom, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. “Cross-Linguistic Data Formats, Advancing Data Sharing and Re-Use in Comparative Linguistics.” Scientific Data 5 (1). https://doi.org/10.1038/sdata.2018.205.

Gautam, Bhojraj, Mark Donohue, and Madhav Pokharel. 2013. Kusunda Linguistics. Canberra: Australian National University. http://kusunda.linguistics.anu.edu.au/wordlist/.

Hodgson, Brian Houghton. 1848a. “On the Chepang and Kusunda Tribes of Nepal.” JASB 17 (1848): 650–58.

Johann-Mattis List, Simon J. Greenhill, Tiago Tresoldi and Robert Forkel. 2019. “Lingpy/Lingpy: A Python Library for Quantitative Tasks in Historical Linguistics. Version 2.6.5” Jena: Max Planck Institute for the Science of Human History. URL: https://lingpy.org

Johann-Mattis List, Christoph Rzymski, Simon J Greenhill, Nathanael E. Schweikhard, Kristina Pianykh, Annika Tjuka, Mei-Shin Wu, and Robert Forkel. 2020. “Concepticon. Version 2.3.0.” Jena: Max Planck Institute for the Science of Human History. URL:https://concepticon.clld.org.

List, Johann-Mattis. 2017. “A Web-Based Interactive Tool for Creating, Inspecting, Editing, and Publishing Etymological Datasets.” In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. System Demonstrations, 9–12. Valencia: Association for Computational Linguistics. http://edictor.digling.org.

Johann-Mattis List, Mary Walworth, Simon J Greenhill, Tiago Tresoldi, and Robert Forkel. 2018. “Sequence comparison in computational historical linguistics.” Journal of Language Evolution 3 (2): 130–44. https://doi.org/10.1093/jole/lzy006.

Rana, B.K. 2002. “New Materials on the Kusunda Language.” Cambridge MA, USA: Fourth Round Table International Conference on Ethnogenesis of South; Central Asia; Harvard University.

Reinhard, Johan, and Sueyoshi Toba. 1970. A Preliminary Linguistic Analysis and Vocabulary of the Kusunda Language. Kathmandu: Summer Institute of Linguistics; Tribhuvan University.

Sagart, Laurent, Guillaume Jacques, Yunfan Lai, Robin Ryder, Valentin Thouzeau, Simon J. Greenhill, and Johann-Mattis List. 2019. “Dated Language Phylogenies Shed Light on the Ancestry of Sino-Tibetan.” Proceedings of the National Academy of Science of the United States of America 116: 10317–22. DOI: 10.1073/pnas.1817972116.

Swadesh, Morris. 1952. “Lexico-Statistic Dating of Prehistoric Ethnic Contacts. With Special Reference to North American Indians and Eskimos.” Proceedings of the American Philosophical Society 96 (4): 452–63.

Cite this article as: Mei-Shin Wu, "New Lexical Data for the Kusunda Language," in Computer-Assisted Language Comparison in Practice, 20/04/2020, https://calc.hypotheses.org/2446.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Concept Similarity in STARLING

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

STARLING is a software package, originally created by Sergej A. Starostin, which is designed for historical linguists who want to build their own etymological dictionaries. It is not only a database system that allows its users to set up a very straightforward relational database structure, but also a package full of surprises, since it contains many methods that are supposed to automate specific tasks in historical linguistics. These range from phylogenetic tree reconstruction via the preliminary identification of sound correspondences up to the comparison of elicitation glosses for their semantic similarity. While phylogenetic reconstruction and sound correspondences are now quite successfully handled in alternative software packages, I thought it would be interesting to discuss the routine for assessing concept similarity in more detail, since it offers interesting possibilities for those who practice historical language comparison.

STARLING, originally created by Sergei A. Starostin (1953-2005, see Starostin 2007[1993] for details on the origins of the software), is one of the oldest software packages devoted to computer-based approaches in historical linguistics. Despite the fact that STARLING has been around for a very long time, not many historical linguists seem to know the system very well. This may be in part to the fact that the origins of STARLING (as far as I know) go back to the early 1990s, a time when specifically scholars in historical linguistics worshipped books or personal collections of excerpts (such as reported by Gabelentz 1891 or Swadesh 1963 more than computers, who would often fail to display special characters correctly.

For me personally, STARLING was the first computer program I was really exposed to. I started to use the software when I was still planning to make my PhD in Sinology, and intended to collect a larger database of Chinese characters along with their readings and the like. Later, I was fascinated by the possibilities which STARLING offered in order to infer phylogenetic trees from lexicostatistic datasets (although I never found a proper description of the algorithm that was used there, I only know it cannot be Neighbor-Joining by Saitou and Nei from 1987, since the trees in STARLING are rooted and dated). Many features in STARLING have directly influenced the design of my own

Computer-Assisted Language Comparison in Practice

Volume 3, Number 5
https://calc.hypotheses.org/2465/

LingPy software package for quantitative tasks in historical linguistics, which is now a larger collaborative project, available in version 2.6.5 (List et al. 2019).

While the data structure has by now diverged quite a bit from the one employed by STARLING, many concepts can still be found in LingPy's basic functions for the manipulation of wordlists. As an example, the function Wordlist.get_etymdict is equivalent to the procedure by which STARLING creates an initial "etymological dictionary" from a larger number of lexicostatistical wordlists which are annotated for cognacy. Also the fact that cognate sets are represented by integer numbers in LingPy and also in EDICTOR, my attempt to offer a similarly convenient way to annotate cognates as provided by STARLING, is due to the influence from STARLING's cognate annotation practice.

In addition to these aspects mainly devoted to data storage and data annotation, STARLING also offers some rather complex computational approaches that help to tackle problems of cognate detection. Unfortunately, not many people have ever heard of these methods, since not many people tried to understand the system in all its power, and most of the descriptions of the methods can only be found in the manual (which is deeply hidden in the software files and difficult to access if one hasn't succeded in installing the software), or in text books, which Sergej A. Starostin co-authored (such as, for example, Burlak and Starostin 2001).

In this context, there are two interesting methods that I want to discuss briefly, one only quickly, and the other method in a bit more detail. They are described in pages 270-275 in the introductory text book by Burlak and Starostin (2005), and treat the identification of regular sound correspondences and etymologically similar meanings.

The algorithm for the detection of sound correspondences is described as follows (ibid. p. 271, my translation):

- a Count the frequency of every phoneme in the list of each of the languages which shall be compared;
- b Take the subset of words N of language A which contain phoneme x;
- c Take the subset of words N' of language B which happen to be translations of the words of subset N of language A;
- d Count the frequency of each phoneme of language B in subset N' and compare it with the general frequency of the given phoneme in the whole list of words for language B;
- e Phoneme x', whose frequency significantly (e.g. according to the three sigma rule) increases the general frequency of the given phoneme, is judged to be corresponding to phoneme x in language A.

What is remarkable is not the description of the procedure itself (the automated identification of sound correspondences was already discussed earlier in the 1990s, as

shown by Guy 1994), but the fact that this procedure was readily implemented in software and could readily be used by linguists already with considerably early versions of STARLING.

The same applies to the second method, which I promised to introduce in the title of this post: the method to identify meanings which could be etymologically similar. Here, the information we are given in Burlak and Starostin is even sparser than for the determination of regular sound correspondences:

Meanings are judged to be similar if there is a root of a proto-language whose reflexes in the daughter- languages happen to have these meanings. The list of similar meanings as well as the list of similar sounds are stored in a special file and can be changed easily. (Burlak and Starostin 2005: 272, my translation)

In fact, if one checks the STARLING software, one can find a file shipped with STARLING, called SENSE.DBF, which consists of two columns, one providing an English headword, and the other providing semantic items. We can find this described in the user manual of STARLING, in the file fSemantic.htm:

SENSE.DBF is a collection of about 7000 English headwords described in terms of their semantic “attributes” or “constituents” (all in all around 400). All the data was extracted from existing etymological computer databases. A record like (HEADWORD) require (V) (ITEMS) to want;to search;to be;able means that in several cases the meaning “require” was associated with semantic “primitives” “to want”, “to search”, “to be” and “able”.

The following screenshot shows how this file looks in the STARLING version I have currently installed on my computer.

C:\StarSoft\bin\sense.dbf

HEADWORD	ITEMS
admit (V)	to think;
admonish (V)	to drive;
admonition	to speak;
adopted parents	relative;
adorn (V)	beautiful;ornament;to know;
adorned with jewels	ornament;
adornment	cloth;
adultery	to play;
advance (V)	new;to go;
adverse	bad;
advice	to speak;
advise (V)	to speak;to drive;
adze	knife;to beat;

Figure 1: SENSE.DBF as it is represented in the STARLING system.

If one counts the element (STARLING makes it easy to export any database into plain CSV format in UTF-8 encoding), one can see that there are as many as 7048 headwords, and 424 different items. In order to determine similarity between the headwords, one can think of a bipartite graph, that is, a graph that has two different types of nodes, one node type in our case reflecting the headwords, and the other type reflecting individual items. Links can only be made between different node types, and each row in the file SENSE.DBF describes individual links from the headword to each of the items.

In order to search for similar words according to this collection, all one has to do is to search for words with similar semantic constituents and apply some criteria as to the number of constituents which two headwords should share in order to be judged as being “similar”.

In order make the similarity functions available inside STARLING available in other software projects, I have written a small Python library for the manipulation of semantic data in linguistics, which I decided to call pysen, as a short form for pysense, and because this reflects a typical German pronunciation of the word “Python”.

In order to install the library, you need to clone it with git for the time being, as it has not yet been officially released:

```
$ git clone https://github.com/lingpy/pysen.git
```

The easiest way to install the library is to use pip:

```
$ pip install -e pysen
```

Once this is done, you can directly test the semantic comparison based on STARLING's SENSE.DBF.

```
>>> from pysen.sense import Sense
>>> sen = Sense()
>>> sen.similar('hand')
 [['hand', 'arm', 's:bone; s:foot; s:hand', 3],
['hand', 'shin-bone', 's:bone; s:foot; s:hand', 3],
['hand', 'calf of leg', 's:bone; s:foot; s:hand', 3],
['hand', 'handful', 's:hand; s:handle', 2],
['hand', 'thigh', 's:bone; s:foot', 2]]
```

The output is a two-dimensional list consisting of four items each. The first item repeats the headword, since I applied some general operations to "smothen" the lookup, which may result in multiple headwords being called when typing one word alone, as you can easily see when searching for similar words for "ear".

```
>>> sen.similar('ear')
[['ear 1', 'hear (V)', 's:ear; s:to hear', 2],
['ear 2', 'jujube', 's:fruit; s:thorn', 2],
['ear 2', 'bush', 's:fruit; s:thorn', 2],
['ear 2', 'ear of grain', 's:fruit; s:thorn', 2],
['ear 2', 'hair', 's:skin; s:thorn', 2]]
```

The second element is the headword judged to be similar by this criterion, the third element provides the semantic attributes, and the last element shows the attribute overlap between the two headwords.

Given that this approach is so straightforward, I could not resist to write a JavaScript equivalent that allows to search interactively for potentially interesting similar meanings. This interface is available at https://digling.org/sense/ and can be invoked by typing any

concept in the field one may think of. If no results are returned, this means that no matches could be found for the respective headword, but obvious concepts, such as, for example, thos from Swadesh's list of 200 items (Swadesh 1952) will all be there, as can be seen in the following screenshot showing matches for "head".

SENSE.DBF Lookup

head

Headword 1	Headword 2	Senses	Matches
head	horn	s:hair; s:head; s:top; s:horn	4
head	forehead	s:head; s:top; s:horn; s:face	4
head	eyebrow	s:hair; s:head; s:face	3
head	beak	s:head; s:face; s:neck	3
head	front	s:head; s:top; s:face	3

Figure 2: SENSE.DBF lookup in JavaScript.

I have been reflecting a long time about this approach, and I like the idea to search for semantically similar words by means of attributes, as this is a flexible system that could serve as a counterpart to searching for similar meanings based on colexification studies (Rzymski et al. 2020). What I have not figured out so far, however, is how Starostin arrived at this list and the attributes. In the description in Burlak and Starostin (2005), they talk about words that have been shown to stem from a common root in a proto-language. This makes of course good sense, as it reflects linguists' intuitive judgments about semantic similarity and plausible pathways of semantic change. However, the 424 semantic attributes in the file SENSE.DBF do not seem to reflect etymological roots, so it is not clear how the dataset was initially created, and I could not find any additional explanation on it in the literature.

In any case, what I find fascinating about this approach is that it provides historical linguists with an alternative to colexification networks when searching for cognates across different meanings, and I would wich that more linguists would think along these directions and help us to improve our knowledge about plausible and less plausible semantics in our etymologies.

References

Burlak, Svetlana Anatolévna and Starostin, Sergej A. (2001): Vvedenie v lingvističeskuju komparativistiku [Introduction to comparative linguistics]. Moscow:Editorial URSS.

Burlak, Svetlana Anatolévna and Starostin, Sergej Anatolévic (2005): Sravnitelno-istoričeskoe jazykoznanie [Comparative-historical linguistics]. Moscow:Akademia.

Gabelentz, Hans Georg C. (1891): Die Sprachwissenschaft. Ihre Aufgaben, Methoden und bisherigen Ergebnisse. Leipzig:T. O. Weigel.

Jacques B. M. Guy (1994): An algorithm for identifying cognates in bilingual wordlists and its applicability to machine translation. Journal of Quantitative Linguistics 1.1. 35-42.

List, Johann-Mattis and Greenhill, Simon and Tresoldi, Tiago and Forkel, Robert (2019): LingPy. A Python library for quantitative tasks in historical linguistics. Version 2.6.5. Max Planck Institute for the Science of Human History. Jena: http://lingpy.org.

Rzymski, Christoph and Tiago Tresoldi and Simon Greenhill and Mei-Shin Wu and Nathanael E. Schweikhard and Maria Koptjevskaja-Tamm and Volker Gast and Timotheus A. Bodt and Abbie Hantgan and Gereon A. Kaiping and Sophie Chang and Yunfan Lai and Natalia Morozova and Heini Arjava and Nataliia Hubler and Ezequiel Koile and Steve Pepper and Mariann Proos and Briana Van Epps and Ingrid Blanco and Carolin Hundt and Sergei Monakhov and Kristina Pianykh and Sallona Ramesh and Russell D. Gray and Robert Forkel and List, Johann-Mattis (2020): The Database of Cross-Linguistic Colexifications, reproducible analysis of cross- linguistic polysemies. Scientific Data 7.13. 1-12.

Saitou, N. and Nei, M. (1987): The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4.4. 406-425.

Starostin, Sergej A. (2007): Rabochaya sreda dlja lingvista [A workplace for linguists]. In: : S. A. Starostin: Trudy po jazykoznaniju [S. A. Starostin: Works on linguistics. Moscow:Languages of Slavic Cultures. 481-496.

Swadesh, Morris (1952): Lexico-statistic dating of prehistoric ethnic contacts. With special reference to North American Indians and Eskimos. Proceedings of the American Philosophical Society 96.4. 452-463.

Morris Swadesh (1963): A Punchcard System of Cognate Hunting. International Journal of American Linguistics 29.3. 283-288.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Why Tag Markup may be Useful for Lexical Data

Ilia Chechuro
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

In lexicography, there are two commonly used types of semantic data categorization: *semantic domains* and *semantic labels*. The difference between the two approaches is simple: semantic domains presume that each lexical meaning belongs to one and only one group (or sub-group of a larger group). Semantic labels do not have such limitations.

Most recently published cross-linguistic wordlist data are often organized into semantic domains, cf. the CLLD-friendly lists, including WOLD (Haspelmath and Tadmor 2009), IDS (Key and Comrie 2005), LexCauc (Belyaev and Forker, unpublished) and others. The same type of data organization has also been used in earlier works, such as Kibrik and Kodzasov (1988, 1990) thesauri of East-Caucasian verbs and nouns or SIL Comparative African Wordlist by Snider and Roberts (2006). The WOLD semantic categorization has also been adopted in the Concepticon project (List et al. 2020) that maps different wordlists using a set of clearly defined meta-concepts to which the entries in the lists are linked.

Below I provide two examples of such annotation from Haspelmath and Tadmor (2009) and from Kibrik and Kodzasov (1990):

(1) WOLD example:
Meaning 5.7: *the potato*
Description:
Typical context:
Semantic field: Food and Drink

(2) Kibrik and Kodzasov (1990) Example:
I. Тело (человека, животного) [Body (human, animal)]
I.1. Голова и шея [Head and neck]
1. *Голова* [Head]: <words for 'head' in different languages>

Computer-Assisted Language Comparison in Practice

Volume 3, Number 6
https://calc.hypotheses.org/2476

This type of categorization, however, may lead to peculiar results. For example, in the WOLD database, the words for *potato*, *olive* and *pepper* belong to 'food and drink', while *pumpkin*, *mushroom* and *banana* are 'agriculture and vegetation'. The structure of the database does not allow to get all the six words with a single query using semantic annotation: one may only get all the words belonging to these categories, including e.g. *farmer* and *field*, which obviously do not belong to 'food and drink'. Mathematically speaking, this structure only allows for a set union operation, but does not allow for a set intersection (simply because the sets do not intersect).

This problem is relevant for most of the wordlists because objects and actions usually belong to multiple domains, just like *potato* is both 'food' and 'plant', *farmer* belongs to 'agriculture and vegetation' (a farmer cultivates plants) and simultaneously is a type of 'social relation' (a farmer has a certain position in the social hierarchy), while *field* belongs both to 'agriculture and vegetation' (because it is where the plants are being cultivated) and 'spatial relations' (because it is a place). In the WOLD type of classification, however, one is always forced to decide, which single semantic domain each word belongs to, even though more than one may be relevant. Thus, if one wants to make a claim about borrowability or other properties of particular semantic domains, domain-based classification may be inapplicable, not to say useless, and the data may require a lot of additional markup and re-annotation.

Traditional dictionaries, on the other hand, rarely use this type of categorization and usually rely on labels. Each word in these dictionaries can be annotated with multiple categorical labels based on its semantics and other properties. For example, in Oxford Learner's Dictionary the word *chap* is simultaneously marked as *British English*, *informal* and *old-fashioned*, something completely impossible in the domain-based approach:

(3) Oxford Learner's Dictionary Example:
Chap **noun**
/tʃæp/
(British English, informal, old-fashioned)

Before computational approaches have been widely introduced to linguistics, semantic domains had a major advantage: searching for data and analyzing them was much simpler when working with lexical data organized into semantic domains. If one wanted to make a statement about lexical processes (e.g. borrowing) and somehow involve semantics, domain-based structure was much more convenient because a linguist would be able to use a pre-existing classification rather than manually look through the whole dictionary looking for each word with a specific label.

With today's technology, however, it has become easier to search and subset data by labels. Since most of the data are stored in digital table formats (e.g. *csv*, *tsv*, *csvw*, *xls*,

etc.) and not on paper, label-based annotation no longer causes problems for searching: one may simply filter the data set using an “IF x IN y” statement.

Thanks to this major simplification, instead of assigning one category per lexeme, one may list the categories it belongs to in a corresponding cell of the column where semantic attribution is stored, similarly to how hash tags are used in Twitter or Instagram. The WOLD categorization could be improved as follows:

Lexical Meaning	**Semantic Tags**	**Lexical Meaning**	**Semantic Tags**
to eat	food and drink	*pumpkin*	agriculture and vegetation; food and drink
food	food and drink	*mushroom*	agriculture and vegetation; food and drink
potato	agriculture and vegetation; food and drink	*banana*	agriculture and vegetation; food and drink
olive	agriculture and vegetation; food and drink	*farmer*	agriculture and vegetation; social and political relations; basic actions and technology
pepper	agriculture and vegetation; food and drink	*field*	agriculture and vegetation; the physical world; spatial relations

We thus allow semantic domains to intersect and assign the words for *potato*, *olive*, *pepper*, *pumpkin*, *mushroom* and *banana* to all categories that seem to be relevant. A query for ‘food and drink’ will result in everything that is related to eating and drinking and the query for ‘agriculture and vegetation’ will result in all plants and everything else related to agriculture. To compare, currently used domain-based structure will yield *some* of the things that can be eaten and *some* of the things that can be planted, leading to obvious difficulties in the analysis.

A similar approach has already been implemented by several scholars in their annotation of wordlist data. One such example is a meta-wordlist by Starostin (2000), supposedly used by Sergei A. Starostin to determine semantic matches between words for cognate detection. Each gloss in the list was assigned a set of semantic tags. To my knowledge, Starostin’s idea was to compare two glosses by their “senses”, i.e. by the lists of tags assigned to each concept. The advantages of this approach are obvious: using Starostin’s annotation one may automatically track cognates by partial semantic matching, which significantly speeds up cognate detection.

This system worked somehow differently from the one proposed here, mainly because of its different purposes. Starostin did not use semantic categorization but assigned what appears to be primitive meanings to the entries, e.g. ‘food’ was tagged with “to eat; grain; fat; belly; fruit”. Searching for similar words in Starostin’s system thus implied simply searching for words with similar sets of semantic tags and deciding how many and which tags two entries have to share in order to be ‘similar’. The STARLING similarity judging

system has been revived in the *pysen* Python library recently published by List (for details see List's previous post in this blog on Concept similarity in STARLING).

The system that I propose in this paper is different from Starostin's in that an 'X' would have to be a type of 'Y' to be assigned this tag, so since 'food' is not a type of 'grain', it could not be tagged with this category. The 'grain' tag could be used for meanings 'wheat', 'oats', 'millet' and the like. Importantly, the two approaches do not exclude each other and can successfully work together. Since Starostin's tagging system is already there and does not require any reworking, it can be treated as a parallel layer of annotation.

The approach I propose here can be implemented in a relatively easy way: the annotation system can be constructed by aggregating the existing categorical annotations from different lexical databases. For example, many lists linked to Concepticon provide a semantic categorization, which can easily be transformed into tag annotation by assigning all the possible categories from the donor lists to the Concepticon entries. Additionally, by linking a system with tags, such as the SENSE database of the STARLING software package to Concepticon, one could use these annotations as metadata for Concepticon concept sets. The suggested implementation is imperfect in many respects, but it could still work as a temporary solution for improving the semantic searchability of Concepticon. A perfect solution would of course be to annotate the database manually, but given the size of the database, it would require a significant amount of work just to develop the tagging system, let alone the annotation itself.

To sum up, when using the label-based approach to semantic classification, linguists are no longer forced to lose information in their annotation. By (re-)introducing label-based categorization to cross-linguistic lexicography and combining it with modern data formats, we are able to take the best from both approaches and create a markup that is both flexible and meaningful as well as easily searchable.

References

Haspelmath, Martin & Tadmor, Uri (2009): World Loanword Database. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: https://wold.clld.org.

Key, Mary Ritchie & Bernard Comrie (2015): The Intercontinental Dictionary Series. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: http://ids.clld.org.

List, Johann-Mattis & Rzymski, Christoph & Greenhill, Simon & Schweikhard, Nathanael & Pianykh, Kristina & Tjuka, Annika & Wu, Mei-Shin & Forkel, Robert (2020): Concepticon 2.3.0. Jena: Max Planck Institute for the Science of Human History. URL: http://concepticon.clld.org.

Roberts, J., & Snider, K. (2006): SIL comparative African wordlist (SILCAWL). Dallas: SIL International.

Starostin, Sergei A. (2000): The STARLING database program. Moscow: RGGU. URL: https://starling.rinet.ru/program.php?lan=en.

Kibrik, A. E., & Kodzasov, S. V. (1988): Sopostavitel'noe izučenie dagestanskix jazykov. Glagol [A comparative study of Daghestanian languages. The verb]. Moscow: Lomonosov University Publishing.

Kibrik A. E. & Kodzasov S. V. (1990): Sopostavitel'noe izučenie dagestanskix jazykov. Imja. Fonetika [A comparative study of Daghestanian languages: Noun. Phonetics]. Moscow: Lomonosov University Publishing.

Cite this article as: Ilia Chechuro, "Why Tag Markup may be Useful for Lexical Data," in Computer-Assisted Language Comparison in Practice, 03/06/2020, https://calc.hypotheses.org/2476.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

A Model of Distinctive Features for Computer-Assisted Language Comparison

Tiago Tresoldi
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

This post introduces a model of segmental/distinctive features for the symbolic representation of sounds, covering almost 600 segments from CLTS (List et al., 2019) mapped to unique sets of bivalent features. It is being designed as an alternative input to vectors of presence/absence built from BIPA descriptors, analogous to other feature matrices like the one by Phoible (Moran & McCloy, 2019). While still under development, it can already be used both for training models of machine learning and statistics, notably decision trees, and for bootstrapping language- and process-specific models, aided by an "universal" and concise reference. The complete matrix is available on Zenodo. Asupporting Python library, distfeat, is available on PyPI.

Background

Syllables and phonemes are the most frequent means for describing phonological entities. While the former are concrete, the latter are more of an abstract notion, arising from the principle of acoustic differences interpreted as contrastive, generally by the test of minimal pair identification. In an often repeated maxim, phonemes are convenient fictions (Ladefoged & Maddieson, 1996).

Just as fictional and convenient is the concept of "features", underlying characteristics that contrast and group speech sounds through "traits" of articulatory or acoustic nature, related to matters like airflow, tongue placement, and vocal cord vibration. The most frequent set of features, also because of a higher "concreteness", are the "descriptors" of the International Phonetic Alphabet (IPA), where a sound such as /tɬ/ is defined as "voiceless", "alveolar", "lateral", and "affricate". While suitable for many analyzes, this phonetic model can get in the way for a symbolic manipulation for typological and historical research. Some features are exclusive (like `palatal` and `palatoalveolar`), some are continuous (like degrees of phonation), some are implied (larynx usage in voiced

Computer-Assisted Language Comparison in Practice

Volume 3, Number 6
https://calc.hypotheses.org/2485

consonants). Similar sounds, such alveolars and dentals, end up having the same overlapping features as less related ones, like bilabials and epiglottals, and a radical separation exists between vowels and consonants. As a result, some processes require complex statements (like suprasegmental assimilations) and known we conceal affinities (such as between retroflex consonants and open back vowels).

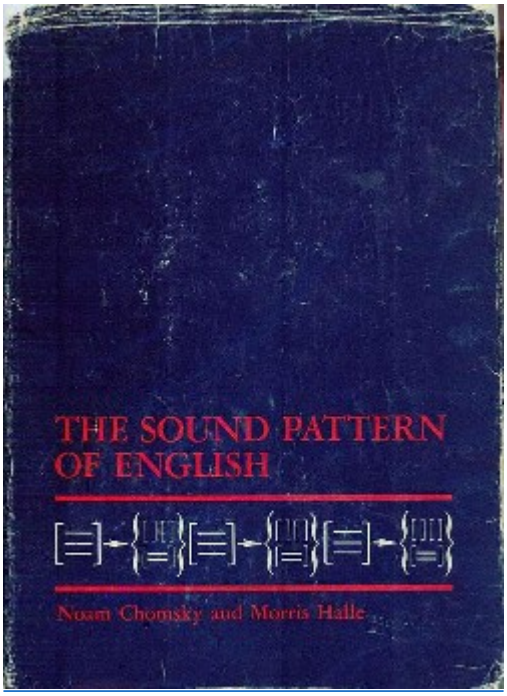

Figure 1: Cover of the first edition of "The Sound Pattern of English"

Segmental features (or, in a more specific context, "distinctive features") are alternative descriptors that focus on representing psychological entities of acoustic-articulatory basis, linking cognitive representations of sounds to their effective manifestations (Hall, 2007). By broadening the contrastive principle, Trubetzkoy (1939) first proposed them in a scheme of different oppositions, such as bilateral, multilateral, privative (or binary), and gradual. Other linguists of the Prague school, especially Jakobson, developed such oppositions, adopting a system composed of binary ones. The proposed collections of around a dozen features in their turn laid the groundwork for Generative Phonology, in which natural classes were designed in line with first-order logic. The most influential product of this school, "The Sound Pattern of English" or "SPE" (Chomsky & Halle, 1968), started a tradition still valued even in dissenting schools, with features such as [sonorant] (marking a periodic low frequency energy) and [delayed release] (expressing a delayed onset of other features).

New proposals were and continue to be developed, usually considering other speech systems (as SPE concerns the sound patterns of English). "Global" schemes are promoted from time to time, but can be of reduced symbolic use either for requiring numerous features or because they are more concerned with abstract models. After all, a universal

reference entails a universality in processes that moves against most of the prevailing theoretical stances, and it does not help that some proposals admit no limits when seeking to fit aberrant cases in a universal pattern (even including reconstructed languages) — also in this case, we might benefit from thinking about the difference between p-linguistics and g-linguistics. In this sense, it is worth remembering Mielke (2008), who investigated the innateness and universality of features in a cross-linguistic database, concluding that they are learned along with language and that in many languages we observe processes better explained by "unnatural" classes. Another interesting innovation in distinctive features are schemes that shift from the monovalence of Jakobson and Chomsky & Hall, advancing bivalent models where, in line with three-value logic, features can be "negative" (-1 or False), "positive" (+1 or True) or undetermined (0 or Null), as the one here introduced (but see, opposed to this practice, Frish, 1996).

The model under development

Feature models are destined for concrete studies, and, as remarked, universal models presuppose a universality that makes it difficult to establish the most economical accounts of actual processes. Nonetheless, a model that uniquely defines the majority of sounds can be useful for the symbolic manipulation as a starting point for compiling specific models using a finished and coherent reference, and it is an inescapable need, like sound classes, when doing cross-linguistic diachronic research. This is the case of Hartman's (2003) strategy for historical reconstruction, for example: although not strictly generativist and involving languages other than English, his system benefits from a development of SPE, allowing him to handle sound sequences through a formal model accessible to its audience and more effective than simple graphemes or IPA descriptors.

For two different project I needed such kind of "common" design. None of the options were entirely satisfactory. Proposals were too complex, too distant from the prevailing linguistic background (an obstacle for collaboration), or excluded entire sets of sounds (such as clicks, alveolopalatals, or rounded labials). More problematic, few cases gave an explicit list of sounds with all the marked features: it is common to find statements in prose that fly over a series of questions, requiring to be "reimplemented" or "reverse engineered" for computational treatment.

The demands were simple: a reduced system that detailed all values for the largest amount of CLTS sounds, to be used as a default in studies or to serve as a guideline when setting up alternative models. Giving up the pretense of mirroring unfathomable psychological entities, the key goal was to aid sound class identification and to offer instruments for similarity assessments. This is illustrated by the algebraic principle that should underlie many decisions: for example, while we can criticize it on a range of phonetic, phonological, and historical grounds, an equation such as "alveolar + palatal =

alveolopalatal" should be roughly accurate for language comparison. This involved a proposal motivated by precepts of least surprise and transparency, conservative in the suggested features and where feature relationships that can be replaced, integrated, or rejected.

The model under development adopts a geometry feature simplified in the picture below, building up on the ones defined in Hall (2007). It expresses 589 of the about 1000 sounds of CLTS through 30 features, encompassing most necessary sounds. Missing segments are entries such as tones and marks, relative length measurements (such as "ultra-long" as opposed to "long"), phonation details (such as creaky-voice and unreleased stops), aliases and sounds considered equivalent (such as "devoiced voiced" consonants, paired to "voiceless" ones), and diphthongs (treated as two separate segments). As its primary reference, it "presupposes that […] features are arranged in a feature tree", integrating and seeking to accommodate different ideas and analyzes, chiefly of SPE, but likewise from Halle & Stevens (1971), Halle & Clements (1983), Sagey (1986), Clements (1985), McCarthy (1988, 1994), Lombardi (1991), Odden (1991), Blevins (1994), Kehrein (2002), and Moran & McCloy (2019).

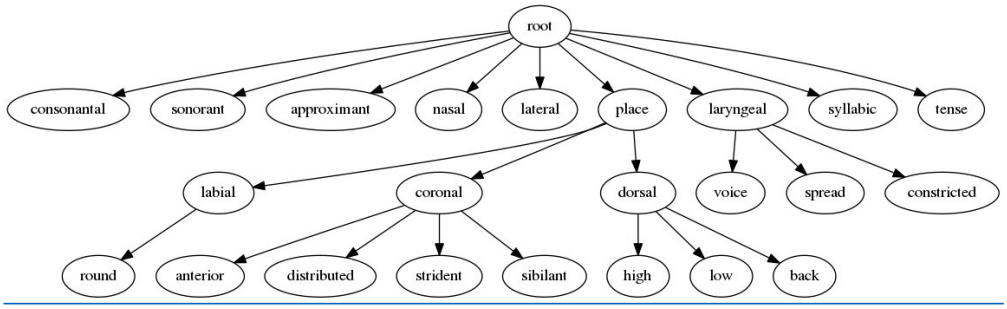

Figure 2: Feature geometry.

A comprehensive characterization of the design's decisions would involve a technicality and an extension not suitable for a post. As the full matrix is available, experts can meanwhile investigate such factors directly, allowing me to only explore fundamental attributes and possibly unexpected factors in this site.

Manner of articulation is expressed by five major features, as in the following table.

	stops	fricatives	affricates	nasals	laterals	rhotics	glides	vowels	clicks
continuant	–	+	–	–	–	+	+	+	–
sonorant	–	–	–	+	+	+	+	+	–
approximant	–	–	–	–	+	+	+	+	–
strident	–	+	+	–	–	–	–	–	–
click	–	–	–	–	–	–	–	–	+

The difference between stops, aspirated and ejectives is given by means of children of the "laryngeal" node.

	p t k	pʰ tʰ kʰ	pʼ tʼ kʼ	b d g	bʱ dʱ gʱ	ɓ ɗ ɠ
voice	–	–	–	+	+	+
spread	–	+	–	–	+	–
constricted	–	–	+	–	–	+

Place of articulation is largely specified by four non-exclusive supra-features: labial, coronal, dorsal, and pharyngeal. The model follows Articulator Theory instead of the Place of Articulation Theory (adopted, for example, in the SPE; the base is McCarthy, 1994). Note that the feature [round] is not identical to the [labial] one, but takes it as an upper node, accounting for issues such protruded and compressed rounding.

More than the schema of Hall (2007), the vocal framework of this model follows Sagey (1986) in spirit, but accepts Hume (1992) arguments for marking front vowels as coronals and all other vowels as dorsals. We can streamline the vocal trapeze in the following table. Note that schwa is undefined, and therefore not displayed in the table below, that it uses the disputed [tense] feature as a purely phonological one, and that rhotacized vowels, such as /aˠ/, are not currently supported (a deliberate decision, in part following the discussion of Chabot, 2019).

		+ant,-back +round	+ant,-back -round	-ant,-back +round	-ant,-back -round	-ant,+back +round	-ant,+back -round
+high, -low	+tense	i	y	ɨ	ʉ	ɯ	u
+high, -low	-tense	ɪ	ʏ	(ï)	(ÿ)	(ɯ̟)	ʊ
-high, -low	+tense	e	ø	ɘ	ɵ	ɤ	o
-high, -low	-tense	ɛ	œ	ɜ	ɞ	ʌ	ɔ
-high, +low	+tense	a	ɶ	ä	(ɶ̈)	ɑ	ɒ
-high, +low	-tense	æ	(ɶ̟)	ɐ	(ɶ̟̈)	(ɑ̟)	(ɒ̟)

As an illustration of the ease in generating derived models, a number of restrictions could be raised, both from a phonetic and phonological point of view, as to the designation of frontal vowels as coronals (although it is not an innovation of this design). In specific, this choice influences the geometry in use and dispenses with the feature [front] common to most models. It is nonetheless rather straightforward, not only in code but even with a spreadsheet program, to generate a derivative matrix in which all the coronal vowels lose this trait and gain a new feature [front]. Other decisions are not disturbed, also due to the easiness in checking if ambiguities, or even errors such as incompatible geometries, are introduced.

Library

As part of this post, I wrote a simple Python library, distfeat, which allows to access the matrix properties without the boilerplate code that would be identical in any analysis. The library provides some additional functions, such as to single out the minimal set of

features needed to distinguish the members of a group of sounds, and includes other matrices, such as one derived from Phoible, to facilitate experimentation.

There is minimal documentation on the package page on PyPI. The code snippet below illustrates some functionalities it offers:

```
>>> import distfeat
>>> df = distfeat.DistFeat()
>>> df.grapheme2features("a")
{'anterior': True, 'approximant': True, 'back': False, 'click': False, 'consonantal': False,
'constricted': False, 'continuant': True, 'coronal': True, 'distributed': True, 'dorsal': True,
'high': False, 'labial': False, 'laryngeal': True, 'lateral': False, 'long': None, 'low': True,
'nasal': False, 'pharyngeal': None, 'place': True, 'preaspirated': None, 'preglottalized':
None, 'prenasal': None, 'round': None, 'sibilant': False, 'sonorant': True, 'spread': False,
'strident': False, 'syllabic': True, 'tense': True, 'voice': True}
>>> df.grapheme2features("a", vector=True)
[True, True, False, False, False, False, True, True, True, True, False, False, True, False,
None, True, False, None, True, None, None, None, None, False, True, False, False, True,
True, True]
>>> df.features2graphemes({"consonantal": "-", "anterior": "+", "low": "+"})
['a', 'aː', 'ã', 'ãː', 'ă', 'a̯', 'ą', 'æ', 'æː', 'æ̃', 'æ̃ː', 'œ', 'œː', 'œ̃', 'œ̃ː']
>>> print(distfeat.tabulate_matrix(df.minimal_matrix(["t", "d"])))
   constricted    laryngeal    spread    voice
--  -------------  -----------  --------  -------
d   False          True         False     True
t                  False
>>> df.class_features(["t", "d"])
{'anterior': True, 'approximant': False, 'click': False, 'consonantal': True, 'continuant':
False, 'coronal': True, 'distributed': False, 'dorsal': False, 'labial': False, 'lateral': False,
'nasal': False, 'place': True, 'sibilant': False, 'sonorant': False, 'strident': False, 'syllabic':
False, 'tense': False}
```

Conclusion

It is imperative to reinforce that I intend this proposal as a pragmatic model for simplifying automatic manipulation. Despite trying to mirror articulatory and acoustic traits as much as feasible, its underlying purpose is to offer different representations in a single scheme, even when it requires simplifications that would be less acceptable in the study of specific systems. The model does not propose to mimic some system underlying all real languages, but to help in explaining them.

References

Blevins, J. 1994. A place for lateral in the feature geometry. JL 30: 301–348.

Chabot, A., 2019. What's wrong with being a rhotic?. Glossa: a journal of general linguistics, 4(1), p.38. DOI: http://doi.org/10.5334/gjgl.618

Chomsky, N., and Halle, M. 1968. The Sound Pattern of English. New York: Harper & Row.

Clements, G. N. 1985. The geometry of phonological features. Phonology Yearbook 2: 225–252.

Frisch, S. 1996. Similarity and Frequency in Phonology. Ph.D. thesis. Northwestern University.

Hall, T. A. 2007. "Segmental features." In Paul de Lacy, ed., The Cambridge Handbook of Phonology. 311–334. Cambridge: Cambridge University Press.

Halle, M. and K. Stevens. 1971. A note on laryngeal features. Quarterly Progress Report 101. MIT.

Halle, M. and G. N. Clements. 1983. Problem book in phonology. Cambridge, MA, MIT Press.

Hartman, L. 2003. Phono (Version 4.0): Software for Modeling Regular Historical Sound Change". In Actas [del] VIII Simposio Internacional de Comunicación Social, Santiago de Cuba, 20-24 de enero del 2003, I.606-609).

Hayes, B. and F. van Vugt. 2012. Pheatures Speadsheet: User's Manual. Los Angeles, UCLA. Available at https://linguistics.ucla.edu/people/hayes/120a/Pheatures/

Hume, E. 1992. Front vowels, coronal consonants and their interaction in non-linear phonology. Doctoral dissertation, Cornell University.

Jakobson, R., C.G.M. Fant and M. Halle. 1952. Preliminaries to speech analysis: the distinctive features and their correlates. Cambridge, Mass.: MIT Press. (MIT Acoustics Laboratory Technical Report 13.)

Jakobson, R. and M. Halle. 1956. Fundamentals of Language. The Hague: Mouton.

Kehrein, W. 2002. Phonological representation and phonetic phrasing: Affricates and laryngeals. Tubingen, Niemeyer.

Ladefoged, P. and I. Maddieson. 1996. The sounds of the world's languages. Oxford, Blackwell.

List, J.-M., C. Anderson, T. Tresoldi, S. J. Greenhill, C. Rzymski, and R. Forkel. 2019. Cross-Linguistic Transcription Systems (Version v1.2.0). Max Planck Institute for the Science of Human History: Jena

Lombardi, L. 1991. Laryngeal features and laryngeal neutralization. Doctoral dissertation, UMass.

McCarthy, J. J. 1988. Feature geometry and dependency: a review. Phonetica 43: 84–108.

McCarthy, J. J. 1994. The phonetics and phonology of Semitic pharyngeals. In Patricia A. Keating (ed.) Papers in laboratory phonology III: Phonological structure and phonetic form. Cambridge, CUP, pp.191–233.

Mielke, J. 2008. The emergence of distinctive features. Oxford, Oxford University Press.

Moran, S. and D. McCloy (eds.) 2019. PHOIBLE 2.0. Jena: Max Planck Institute for the Science of Human History. (Available online at http://phoible.org, Accessed on 2020-06-17.)

Odden, D. 1991. Vowel geometry. Ph 8: 261–289.

Sagey, E. 1986. The representation of features and relations in nonlinear phonology. Doctoral dissertation, MIT.

Trubetzkoy, N. S. 1939. Grundzuge der Phonologie. Travaux du Cercle Linguistique de Prague 7, Reprinted 1958, Gottingen: Vandenhoek & Ruprecht. Translated into English by C.A.M.Baltaxe 1969 as Principles of Phonology, Berkeley: University of California Press.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

How to do X in linguistics? A new series of blog posts

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

The post introduces a new series of blog posts in the CALCiP blog, devoted to manuals about aspects of scholarly work in linguistics which are not often discussed in the literature.

I cannot remember when I decided to become a linguist. I cannot even remember when I first called myself a linguist (as opposed to a student, a Sinologist, or a scientist). But I can remember when I wrote my first review for a linguistics journal, and I also remember that it came close to a catastrophe, since I maintained a very hostile tone, I didn't like the paper, thought the authors were badly informed, and didn't want to allow the paper to be published.

In the end, the paper was accepted against my rejection, and I moved on to other reviews. By now, I have done enough of them to know that one should never write reviews in a hostile tone, even if one rejects a paper. It's about science after all, and science is about arguments, not about showing that somebody is less smart than oneself.

My first review for a scientific journal was not the first premiere I celebrated during the last 10 years. I learned how to respond to reviews, I learned how to read reviews properly and to keep my calm when reading them, I learned how to write cover letters when submitting papers with multiple authors, I learned how to write grant applications, I learned how to review grant applications, and I also learned how to collaborate with different people.

What all these things I learned have in common is that I learned them chiefly by doing. Nobody ever told me how to write a review, nobody showed me how to respond to a review, nobody explained to me how to write cover letters (until I had to write my first one, I didn't even know what a cover letter is), and nobody gave me tips on collaboration.

When I decided to try to play the game and pursue a career in science, I never knew that being a scientist would involve writing reviews, cover letters, and collaboration

Computer-Assisted Language Comparison in Practice

Volume 3, Number 7
https://calc.hypotheses.org/2501

emails. The first time I heard about a “poster presentation” at a conference, I couldn’t understand what this would refer to, since “posters” were something commercial for me, something you sell, or something you receive along with some magazine you buy, like the posters of pop stars or movies which people pin on their walls. There are so many things one learns indirectly, rather than being officially informed in science, that I can barely remember how I imagined the life of a scientist when I was still a university student.

One can argue that it is natural that there are things that one needs to learn by doing them, rather than being formerly instructed. For example, it would be silly to give young scientists a course in how to become a celebrity, although we have seen quite a few scientists who have used their science for this purpose (which often ends with scientists talking more about themselves or their views on politics and culture than about their actual scientific interests). But there are also quite a few never-taught scientific skills where it would not hurt to find a nice manual on wikiHow.

After discussing with the other fellow contributors of this blog, we have therefore decided to launch a new series of blog posts (which we try to publish in a monthly rhythm) where we explicitly discuss some scientific howtos for which one barely finds advice in handbooks and tutorials. We will do this in a sporadic manner, touching topics that we find while doing science, so this introductory post does not provide any closer plan for the future.

However, I can say so much already: I plan to discuss more closely how to review, how to respond to reviews, how to revise a paper, and also how to cite properly, and the other contributors of this blog have already confirmed that they might have the one or the other topic they would like to share as well.

Furthermore, as this is an open platform, if you want to contribute to this series, or just contribute to another topic that has something to do with the broad topic of computer-assisted language comparison, do not hesitate to get in contact with me. We are always happy to broaden the circle of contributors.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

How to write an initial review for a journal in linguistics? (How to do X in linguistics 1)

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

Writing reviews for a journal is one of those things which most scientists never actively learn. For laypeople, this may be surprising, given how often the scientific method with its rigorous peer review procedure is being mentioned in the news nowadays. How can it be, one may ask oneself, that this procedure that is usually presented as the core principle of scientific reasoning, is never really actively taught? If the review by experts is the core of the scientific method and what decides about the acceptance of an article, how can it be that scientists do never take a course on article reviewing, and how can it be that reviewers are (as I have previously discussed in a German blogpost) themselves never reviewed or graded?

I have no real idea why this is the case. Maybe, it reflects just the way humans behave normally? We are all a bit conservative and reluctant to change. The moment we write our first review, we may shout and complain that there's no real instruction of how to do it. But at the same time, we feel so proud that we were finally asked to write one, that we forget all the problems we had in the moment we submitted our first review. When we then meet younger colleagues who are in the same situation in which we once were a long time ago, we just tend to think, as again many people do: don't complain, young person, I also had to go through this, why should you have it any easier than me?

This situation of not having clear review instructions and not having any clear process by which scholars sit down and evaluate our reviews is very unfortunate, specifically also because writing reviews is a complex business. I realized this again when I sat down to prepare this blogpost, since I could not find a way to prepare a straightforward how-to-guide for writing a review in linguistics, since I do not even know if people would consider my own reviews as useful. So what I will provide in this post is less a full guide

Computer-Assisted Language Comparison in Practice

Volume 3, Number 8
https://calc.hypotheses.org/2504

of writing a review, but more an at times loose collection of ideas and thoughts on the topic of review writing that have been accompanying me during the past years.

What is an initial review for a journal?

Before we start with some concrete suggestions and thoughts, I have to clarify what I mean by an "initial review for a journal", since in science, there are quite a few different kinds of reviews one can write. In contrast to a review article, which summarizes the state of the art in a given research field, a book review, which critically evaluates scientific monographs, and a review for a grant proposal, where one decides whether a scholar should receive funding for a project or not, a peer review for a journal refers to the report a scientist writes on a journal article that was submitted to a journal for publication and which is supposed to help the editors of the journal to assess whether the article merits publication in their journal or not.

I distinguish the initial review from the follow-up review. The initial review is the report one writes upon the first submission of an article. Since articles are often not directly accepted, but rather sent back to the authors with a list of change requests, typically labelled as "minor modifications" (almost accepted, no follow-up review required) and "major modifications" (article has to be reviewed another time), there is a considerable difference between the review one writes upon initial submission and the follow-up review. While the first needs sufficient detail to summarize and assess the paper, the follow-up review may be quite short, even just a sentence at times, provided the authors have convinced the reviewer with their modified manuscript.

In this post, I will share my thoughts regarding initial reviews for a journal. When receiving manuscripts for publication, the editors of a journal select reviewers who can help them to assess the quality of the work, and then invite them via email to share their thoughts. In the following, I will try to run quickly through the major stages of writing an initial review for a journal, thereby discussing (1) what one should consider before accepting to review a paper, (2) initial quality checks of the study, (3) the preparation of the review report, (4) the active writing process, and (5) how to decide on a recommendation.

1 What one should consider before accepting a review invitation

When being asked to review, one should first make some background check on the journal. This is not only helpful in order to find out if the journal is not a predatory publisher, or one of those journals which try to maximize profit by bothering potential reviewers with tons of automated emails (I had the most annoying experience with the Frontiers journals so far, but I am sure there are journals out there who do worse), but also to understand the journal's basic review procedure. Some journals have set up very

fancy review practices and guidelines, which can require a lot of work (Frontiers journals are again an example), some publish all reviews online (MDPI journals are an example here), some require non-anonymous reviews, and some have very explicit forms one needs to fill out. Although there are not many reasons to not make a review for a given journal, it is useful to keep these questions in mind, since it will also influence how much time one will have to devote to the review.

As a second step, one should read the abstract, and make sure that one feels equipped to evaluate the research. If this is not the case, one should politely refuse to review the study and contact the editors, recommending colleagues who could do the review instead.

2 Initial quality checks

After having accepted to review a paper, one needs to organize oneself and decide on which day one wants to write the review (reviews should not take longer than one day, although younger scholars may feel they need a bit more time for this, and it may be helpful to do a review in smaller pieces over several days, in order to avoid that one writes it in a biased mood).

At the same time, when having received the electronic version of the paper, one should make sure that the basic conditions of the review have been met. This means that one should check (1) if data and code are required, they are offered in editable form, so that one can test them on one's own computer, and that one should make sure that (2) there are no conflicts of interest with respect to the study.

If one detects conflicts of interest (e.g. that one works on the same topic and is about to submit a similar paper to another journal), one needs to withdraw the review. To help the editors in finding a good reviewer as replacement, it is always useful to recommend a colleague who could do it. If data and code are not submitted, one should inform the editors and ask them to contact the authors so that they can (1) submit data and code and (2) confirm that data and code will be shared upon publication as well. In order to make sure editors understand the urgency of this claim, it is useful to cite recent studies on open research, such as Nature's (2018) editorial that emphasizes reviewers rights to request data and code during the review process). It is also useful to point editors to the principles of open science and FAIR management of data and code (Wilkinson et al. 2016. Last not least, it helps to point to one's institute's policy of not supporting irreproducible research.

Referring to one's institute's review policy is especially helpful to emphasize the importance of the claim, while at the same time making sure the editors do not blame one for being overly pedantic. The past years have seen an abundance of studies in which data and code were not submitted and in many cases, authors have even refused to share them upon request. One may argue that our research is less harmful when not being substantiated with data and code, since there are no consequences of our findings

(contrary to the missing data in medical research). But if we want to stand up for the principles of scientific research, we need to stand up for the basic principles of transparency in our research. If I start trusting somebody who told me a dated phylogeny for some language family yields 6000 years as divergence time without supplying data and code, what is the difference in trusting somebody who claims to have turned water into wine without providing the chemical formula?

When authors point out that data sharing would expose problems of privacy concerns or similar, it is always possible to find appropriate solutions. Copy-righting issues, for example, can be creatively handled in many cases, by sharing parts of the data. If parts of the data consist in the form that it would impact on the privacy of individuals, scholars can share the data in a maximally anonymized form (see List 2020a for an example on sharing copyrighted rhyme data). If the data are not available completely, due to unclear licenses, one can submit code that allows the readers to crawl the data directly (see the approach in Tjuka et al. 2020 as an example).

All in all, it is important to emphasize that there is no reason to submit research that is irreproducible, and one should refuse to review a study in case editors do not help in this regard or authors do not comply with it. In this way, one will not prevent the study from being published, but one can raise concerns and does not contribute to endorse bad scientific practice.

3 Preparing the review report

There are different strategies one can follow when it comes to prepare the review report. It may, for example, be useful to first read the paper and make an excerpt, in which one copy-pastes quotes along with short comments (similar to the EvoBib collection, cv. List 2020b). But this always depends on the initial impression that a study makes on the reviewer. If the study seems to be good but with a few things that could be enhanced, it is probably best to read it and make some notes to oneself. However, if the study is significantly flawed, and this is already clear from the beginning, it may not be worth to read it in all its details, but rather pick out the most important points to confront the authors.

It may happen to reviewers, especially when they are younger, that they feel intimidated by the overly complex style in which a study is written. As a result, reviewers may endorse the study in order to avoid that they have to admit that they did not really understand it (as it apparently happened as part of the Grievance Studies Affair).

As a rule, reviewers should be honest about their confidence when criticizing a study. This means also, that, when being confronted with a huge bunch of mathematical formulas which one does not understand, one has the right to emphasize that one cannot judge this oneself but recommends the editor to look for a statistician. The same should apply to text that is hardly readable due to an exaggerated use of non-standard

terminology. Additionally, one also has the right to criticize overcomplexity. Scholars should be able to adapt their articles to their readership. At times, one has the impression that formulas and exaggerated terminology are used to intimidate readers rather than to inform them. Good reviewers and editors are needed in order to spot these cases.

While I usually try in my reviews to understand in full what methods and arguments authors have been using, I also often encounter situations where I simply have to admit that a given study goes beyond my expertise. Since I know that editors tend to have huge troubles in finding enough reviewers for their submissions, I try to still write a review in these cases, but I always make it clear to the authors that I only comment on the points where I feel competent, and I repeat this also in a personal letter to the editor, to make clear that my review should not weigh as much as an alternative review by somebody who understands the study in all detail.

4 Writing the review report

Before writing the review report, one should check the journal's online system, as some journals have very specific questions. If this is not the case, one will have to write a report in free form, here it is recommended to follow some basic structure, even if there are no clear guidelines for free reviews.

For an initial review one should always provide a summary of the paper. This helps to illustrate that one has understood the study properly. This summary should then be accompanied by a short recommendation and assessment. Here, most journals distinguish the "magic ABCD", namely (A) accept without modifications, (B) accept with minor modifications, (C) revise and resubmit, and (D) reject. Unless one accepts a study without modifications (which rarely happens anyway), more detailed comments on major issues should follow in a second section, and minor comments (spelling, layout, etc.) should be summarized in an additional section. References should be provided in a final section.

While this structure is by no means required and only stems from my own experience, it is useful for authors to revise their studies afterwards, as it clearly states what is considered as important major revisions, and what can be done quickly in the form of minor fixes.

Not all reviewers provide detailed references for literature they mention, but I consider this as an extremely bad style. Just mentioning that some Shannon said something in 1993 about juggling and science, for example, is bad style, as the authors have no real way to verify it, especially if the names are common. Even if there is only one Shannon who wrote an article on juggling and science in 1993 (it happens to be Claude Shannon, 1916-2001, the father of information theory), the rigor we demand as reviewers from the authors should be the same we demand from ourselves as reviewers writing a review.

It is not always easy to guarantee one's anonymity, as it may be obvious from comments of a reviewer who one is, especially in areas where there are only a few experts. Therefore, one should always write as if the review was openly accessible to anybody, assuming that the authors know one's identity and that colleagues can see one's review along with the article. While this seems to go against the original idea of blind peer review (which is supposed to shelter younger scholars from the revenge of senior scholars when submitting demanding reviews), it has helped me a lot to refrain from being polemic and trying to be constructive instead. While anonymity may protect reviewers, it is not a wildcard to be offensive. Unfortunately, not all colleagues understand this.

5 Deciding on a recommendation

Although the "magic ABCD" of recommendations looks rather straightforward and simple, it may be difficult to decide on a good recommendation. Since there is less of a competition for the best journals in the humanities and editors are usually glad if they can fill their next volume with enough articles of an acceptable quality, it does not happen too often that papers are directly rejected if they fulfill general standards of scientific quality. Similarly, given that "accept without modification" means that the article is judged to be in a state where it could be published tomorrow, it is also unlikely that any study will fulfill these demands upon first submission, and my experience is that all articles which I have submitted in the past have greatly profited from critical peer review. As a result, reviewers will usually pick one of the two lighter options and recommend a resubmission with major revisions or to accept the paper with minor modifications.

Reasons for a full rejection are: (1) extreme hypotheses in a paper that are falsely confirmed (e.g., by wrong methods), (2) claims for new methods that have been developed although these methods already exist, (3) scientific misconduct (e.g., plagiarism), (4) the study appears to be out of topic for the journal.

Papers need a major revision if one feels that (1) whole passages should be rewritten and restated, or (2) that data and code have not been submitted and need to be inspected in a second review round. For minor revisions, where it is not guaranteed that one will see the paper again as a reviewer in revised form, these recommendations can be given when only few things need to be added, like footnotes and references, and the majority of the study seems to be convincing enough.

No matter what recommendation one gives to the editor: unless the paper is a complete disaster, one should always try to encourage the authors, even if one feels that there is still a lot to be done. Even if I recommend to reject a paper, I try to give the authors some suggestions on how they could turn their work into a successful study. A good review, even if it is a rejection, helps the authors to improve.

Final remarks

This collection of ideas on how to write an initial review for a journal has taken me more time than I originally expected. It is probably also less organized than I initially hoped, and I even do not know if it is exhaustive enough to serve as a useful guideline for those who have yet to write their first reviews. However, despite the fact that I am not necessarily content with this contribution, I hope that it could help to contribute to a future debate about reviewing in linguistics in specific and in science in general. The fact that one of the most important aspects of scientific practice is barely discussed, taught, and evaluated should remind us all that the scientific method is a fluid collection of best practices in scientific research, it is not an approach that has reached perfection and does no longer need to be questioned.

References

Johann-Mattis List (2020-08-24): General remarks on rhyming (From rhymes to networks 2). The Genealogical World of Phylogenetic Networks 9.8. .

Nature, Editorial Board (2018): Referees´rights. Nature 560. 409.

Shannon, Claude E. (1993): Some. In: : Claude Elwood Shannon. Collected Papers. New York:IEEE. 850-864.

Tjuka, Annika and Forkel, Robert and List, Johann-Mattis (2020): Linking norms, ratings, and relations of words and concepts across multiple language varieties. PsyArXiv 0.0. 1-24. [Preprint, under review, not peer-reviewed]

Wilkinson, Mark D. and Dumontier, Michel and Aalbersberg, Ilsbrand J. and Appleton, Gabrielle and Axton, Myles and Baak, Arie and Blomberg, Niklas and Boiten, Jan-Willem and da Silva Santos, Luiz B. and Bourne, Philip E. and others (2016): The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

A list of 171 Body Part Concepts

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

The body of most human beings consist of similar parts such as a head, arms, legs, and so on. Many body parts also occur in animals. The shapes and functions of body parts are universal across cultures, but speakers of various languages choose to categorize the body differently. For example, Vietnamese has a single word (*tay*) for the concepts HAND and ARM. The universality of the human body and its categorization into different parts have attracted attention across research areas such as lexical typology and cognitive science. Therefore, I present a comprehensive list of human and animal body part terms based on German which were mapped to the concepts in the Concepticon (List et al. 2020). The list is intended for investigations on cross-linguistic naming patterns of body parts.

Body Parts as the Building Blocks of Cognition

Humans have bodies and they (often) consist of the same parts. Most of us see, feel, or know that a certain body part exists. I see my legs, feel my back touching the chair and know that I have an appendix although I've never noticed it. In addition, many experiences with the world around us are formed through an interaction, perception, or sensation of our body parts. As a German speaker, I pick up my cup of coffee with my **hand**. But as a Vietnamese speaker, I would pick up the cup with my **arm**. The movement is the same so the perception should not differ across the two language speakers. Still, the focus seems to shift from the intricate grasp of the hand and the fingers around the cup to the much bigger movement of the whole arm. Thus, the question arises whether speakers of diverse languages experience their interaction with the world differently? Finding the answer to this question is the motivation for the present blog post. Here, I will establish the basis for a comparison of body parts and their denotation across languages.

Researchers in language documentation and lexical typology underwent great efforts to describe and compare names for body parts in various languages. One of the first cross-linguistic studies of body part nomenclature by Andersen (1978) revealed common principles that speakers use to categorize the body: 1) a hierarchical structure, 2)

perception of spatial alignment, and 3) visual properties. Furthermore, Andersen (1978) proposed the following body parts to be universal in that all languages should have a word for *head*, *eye*, *nose*, and *mouth*. However, Enfield et al. (2006) showed in their large-scale project on the cross-linguistic categorization of the body that this assumption does not hold for all languages. Unsurprisingly, Wierzbicka (2007) defends the view of semantic universals within the body part domain in direct response to Enfield et al. (2006). But whether or not words for certain body parts exist in all languages might not be the most intriguing question. As Brown (2013a, 2013b) illustrated, the variation for colexifications of the concepts HAND and ARM as well as FINGER and HAND seems to follow certain patterns. For example, a geographical cluster in Australia and North America can be found where languages tend to have a single term for FINGER and HAND (the data is available in WALS, Dryer & Haspelmath 2013).

Another view of the study of body parts comes can be found in cognitive science. Semantic knowledge of body parts seems to be deeply rooted in our memory (Majid 2010). Although languages vary in which body parts they denote, all speakers may have a mental representation of the body and its parts. Majid (2010) concludes that the categorization of the body is, on the one hand, based on perceptual constraints. On the other hand, speakers need to learn the linguistic conventions of their language. In a comparison of body part nomenclature across Dutch, Japanese, and Indonesian, Majid (2015) further explored this view and showed that names of body parts often align with visual discontinuities. Nevertheless, embodied representations do not explain the whole picture. This contrasts with the embodiment thesis that human cognition is embodied and shaped by our bodily perception (Wilson 2002, 2011).

Introducing the Body Part List

The brief overview of the literature demonstrates some of the discussions surrounding the study of body part categorization. At first glance, the human body seems to be the perfect starting point for cross-linguistic universals, but the picture turns out to be much more complex. To examine patterns of body part nomenclature on a larger scale, I created a list of human and animal body part terms based on German which were mapped to the concepts in the Concepticon. The list includes 171 body part concepts from ADAM'S APPLE to WRIST. Each concept was categorized into human, animal, or human/animal. In addition, the concepts received tags for gender (male/female) and if applicable, a reference for the relation to other body parts (part of, instance of). The list was published on Zenodo (DOI: 10.5281/zenodo.4058506) and will be available in the next version of Concepticon(List et al. 2020).

The following table shows a small excerpt of the body part list (note that the table was pivoted here for better readability):

CONCEPTICON_ID	802	803	1673	73	1303	3716
CONCEPTICON_GLOSS	ADAM'S APPLE	ANKLE	ARM	BEAK	FINGER	OVARY
CATEGORY	human	human/animal	human/animal	animal	human/animal	human/animal
GENDER	male	male/female	male/female	male/female	male/female	female
GERMAN	Adamsapfel	Knöchel	Arm	Schnabel	Finger	Eierstock
ENGLISH	adam's apple	ankle	arm	beak	finger	ovary
PART_OF			ARMPIT, ELBOW, LOWER ARM, UPPER ARM		FINGERNAIL, FINGERTIP	
INSTANCE_OF					FOREFINGER, LITTLE FINGER, MIDDLE FINGER, THUMB	

References

Andersen, E. S. (1978). Lexical universals of body-part terminology. In Greenberg, J. (Ed.). Universals of human language, Word Structure. Stanford University Press. p. 335-368.

Brown, C. H. (2013a). Hand and Arm. In Dryer, M. S. & Haspelmath, M. (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. Retrieved fromhttp://wals.info/chapter/129

Brown, C. H. (2013b). Finger and Hand. In Dryer, M. S. & Haspelmath, M. (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. Retrieved fromhttp://wals.info/chapter/130

Dryer, M. S., & Haspelmath, M. (2013). The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. Retrieved fromhttp://wals.info

Enfield, N. J., Majid, A., & Van Staden, M. (2006). Cross-linguistic categorisation of the body: Introduction. Language Sciences, 28(2-3), 137-147.

List, J.-M., Rzymski, C., Greenhill, S., Schweikhard, N., Pianykh, K., Tjuka, A., Wu, M.-S., & Forkel, R. (2020). Concepticon 2.3.0. Jena: Max Planck Institute for the Science of Human History. Retrieved from http://concepticon.clld.org

Majid, A. (2010). Words for parts of the body. In Malt, B., Wolff, P., & Wolff, P. M. (Eds.). Words and the mind: How words capture human experience. Oxford University Press. p. 58–71.

Majid, A., & van Staden, M. (2015). Can nomenclature for the body be explained by embodiment theories? Topics in Cognitive Science, 7(4), 570-594.

Wilson, M. (2002). Six views of embodied cognition. Psychonomic Bulletin & Review, 9(4), 625-636.

Wilson, R. A., & Foglia, L. (2011). Embodied cognition. Retrieved from https://plato.stanford.edu/entries/embodied-cognition/

Wierzbicka, A. (2007). Bodies and their parts: An NSM approach to semantic typology. Language Sciences, 29(1), 14-65.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Annotating Rhyme Judgments for a Complex Corpus of Manuscript Sources: Making Sense of the *Cang Jie pian* 蒼頡篇

Christopher J. Foster
Pembroke College
University of Oxford

Establishing a standardized annotation framework for communicating rhyme judgments identified in historical texts will both ease the use of computational tools for rhyme analysis, and hopefully inspire greater collaboration amongst scholars interested in historical linguistics. The framework we have proposed (List, Hill & Foster 2019), was designed with simplicity, exhaustiveness, and flexibility in mind (p. 30), with the intension of eventual inclusion in the Cross-Linguistic Data Formats initiative (https://cldf.clld.org). Further testing of the framework is desired to demonstrate its utility and identify areas requiring refinement. This study presents such a test on rhyming in the *Cang Jie pian* 蒼頡篇, an ancient Chinese scribal treatise only recently reconstructed from a complex corpus of surviving manuscript fragments. In a follow-up study, the proposal will be formally evaluated by providing code to test the annotations.

Although developed initially with a focus on the reconstruction of Old Chinese through an analysis of received ancient Chinese texts, such as the *Shijing* 詩經 (*Classic of Poetry*), the rhyme annotation framework by List, Hill & Foster (2019) might be applied fruitfully to other types of Chinese texts and to different poetic traditions besides. One such example that we raised was the opening chapter of an important Chinese primer, known as the *Cang Jie pian*, newly discovered among various fragments of Han period (206 BCE-220 CE) wood- and bamboo-strip manuscripts (see also Foster 2017; Fukuda 2004; Liang 2015). To continue the testing of our framework, we have compiled a dataset that

Computer-Assisted Language Comparison in Practice

Volume 3, Number 10
https://calc.hypotheses.org/2525

expands upon this example by including the entirety of the *Cang Jie pian*'s content, as it is found on five of our major manuscript witnesses. The present dataset corrects and supplants that for the *Cang Jie pian* linked to in List et al. 2019 (https://doi.org/10.5281/zenodo.3252141).

The *Cang Jie pian* was an important scribal treatise (*shishu* 史書) in early imperial China. Through mastery of this text, students were able to attain coveted government positions. Although the *Cang Jie pian* failed to be transmitted among our received corpus of ancient Chinese texts, manuscript discoveries over the past century shed new light on the nature of its content. Indeed, the *Cang Jie pian* is ubiquitous among recently unearthed caches of Han period manuscripts, extant now in sixteen different collections. Furthermore, it appears in diverse archaeological contexts, from an aristocratic burial in Anhui, to the military installations in Gansu, and even among the remains of the Jingjue 精絕 kingdom in Niya 尼雅, Xinjiang. The Han dynasty witnessed the construction of China's first enduring empire. Accompanying this political reform were linguistic changes and shifting norms in the manuscript culture. The central role played by the *Cang Jie pian* in scribal education, alongside its widespread distribution during the Han dynasty, recommend it as a potentially fruitful source for research into the standardization of Chinese language and writing at this time.

CJP Rhyming Data Documentation

"CJP Rhyming Data" is offered as a first step towards the systematic analysis of the language of the *Cang Jie pian*, to aid in the reconstruction of Han Chinese and the articulation of linguistic changes that occurred during this period. In what follows is any explanation of the information included in this dataset and its presentation. The first sheet, "CJP Rhyming Data," presents a sortable table filled in with the pertinent information from our manuscript sources. Additional explanations for the columns may be found below. The second and third sheets, "Rhyme ID Index" and "Pre-VT & VT Line Index" respectively, help the user to locate and compare data across rhyme IDs, editions and stanzas. The final sheet, "Bibliography," provides standard references for the primary source publications and the secondary scholarship cited in the previous sheets.

In ancient China, texts often were written on scrolls of bound strips made from wood or bamboo strips (Tsien 2013). As mentioned before, content from the *Cang Jie pian* is prolific among recent manuscript discoveries. There are hundreds of individual strips or fragmented strip-pieces which relate to this text. Accounting for the entirety of this data will take a significant amount of time, and may necessitate multiple versions for "CJP Rhyming Data." The eventual goal, for those with specialist sinological interests, is to

include all manuscript evidence in the database, along with every contended transcription and interpretation proposed by paleographers, to support a broader apparatus for the study of the *Cang Jie pian*. For the time being, however, "CJP Rhyming Data" prioritizes data relevant only to the structural rhymes around which the text is based, as a test for our framework, meeting our computational goals.

1. Rhyme Structure in the *Cang Jie pian*

The *Cang Jie pian* is a tightly structured text and is organized around rhymes. Every line is four characters in length, with a rhyme position falling at the conclusion of every second line (e.g., every eighth character). Each chapter, moreover, participates in a single overarching rhyme scheme. Knowing these rules greatly eases our adjudication of where rhyming positions should occur. "CJP Rhyming Data" presents these structural rhymes, and the variants found in those positions. It is, of course, possible that irregular internal rhyming exists in the *Cang Jie pian* as well, or that other interesting linguistic phenomena, such as alliteration, are present. While "CJP Rhyming Data" is not designed to highlight these features, it may be of service in their discovery and eventual analysis.

2. Editions, Sources, and Textual Reconstruction

For our purposes here, we differentiate between three editions of the *Cang Jie pian*:

(1) An early version in which chapters vary in length, but contain over 100 characters. This will be called the "Pre-VT" edition. Witnesses to the Pre-VT edition include the Peking University **Cang Jie pian* (Beijing daxue chutu wenxian yanjiusuo 2015; abbreviated PKU), and the Fuyang Shuanggudui 阜陽雙古堆 **Cang Jie pian* (Hu and Han 1983; Zhongguo jiandu jicheng bianji weiyuanhui 2001+; abbreviated FY).

(2) A later version, said to have been edited by "village teachers 閭里書師," that divides the content into 60 character chapters. This will be called the "VT" edition. The main witness to the VT edition is the so-called "Han board" **Cang Jie pian* (Liu 2019; abbreviated HB), for which we will have further comment below. Also included is JYX EPT 50.1, from among the "new Juyan strips 居延新簡" (Zhang 2016; cache abbreviated JYX). This single bamboo strip writes out a nearly complete version of the "opening chapter" to the *Cang Jie pian*.

(3) An edition based on the VT text, but which adds rhyming commentary to each base line. This will be called the "SQZ" edition, after the only manuscript witness, known as the Shuiquanzi 水泉子 **Cang Jie pian* (Zhang 2015; abbreviated SQZ). Because the SQZ *Cang Jie pian* supplies additional testimony for the VT edition, it is entered into the database twice, as evidence for both the VT and SQZ editions.

These are our main sources, and each has been documented in full in "CJP Rhyming Database." Other manuscripts are included, but only when they bear information relevant to the structural rhymes. The above categorization of our manuscript sources into three *Cang Jie pian* editions is speculative and meant primarily as an heuristic guide. Note, for example, the conflict between the PKU and FY mss in the 漢兼 stanza (lines 3-6), which may be edition-level variation; despite this conflict, we retain both as Pre-VT sources. Furthermore, due to the fragmented state of many of our *Cang Jie pian* manuscripts, it is not always feasible to determine from which edition their content derives. As a general working hypothesis, all manuscripts besides PKU and FY are treated as VT sources.

Comparing the Pre-VT and VT editions of the *Cang Jie pian*, it appears that the VT edition rather mechanically divided the longer chapters of its predecessor into shorter 60-character long segments, without significant further alteration to the content (Foster forthcoming). In other words, the content and line order of Pre-VT and VT largely coincide, even though there are alternative divisions for larger textual units (i.e., chapters). For this reason, despite the fragmentary nature of our sources, we can propose reconstructions that draw across manuscripts witnesses and editions with some measure of confidence. For instance, the placement of FY C046 before PKU 1 in the Pre-VT edition of the text is justified in part because of the evidence found in the VT edition, which on HB 3 has parallel content to that of FY C046 and PKU 1 written consecutively together. A more extensive discussion of the textual history of the *Cang Jie pian* and the methodologies that have been employed in its reconstruction may be found in Foster 2017. Of course, a degree of caution still is warranted when drawing across different manuscript witnesses to reconstruct hypothetical base text and line breaks. (See the note to Pre-VT #2 line a, PKU 65, or to SQZ VT 3 line a, SQZ C072, for examples).

3. Individual Column Specifications

The columns employed in the "CJP Rhyming Data" table follow those proposed in List et al. (2019, see especially p. 31). Further specifications pertinent to this particular case study are as follows. These modifications highlight some of the issues and refinements needed to work with a complex manuscript corpus:

Column ID

Every entry is given a unique numerical ID, allowing for convenient location of data during discussions. Currently, identifiers increment by 1, but this may change when future versions lead to the insertion of new data in the document.

Column POEM

CJP stands for *Cang Jie pian* 蒼頡篇. Non-CJP is listed for materials that do not belong to the text proper, or for which there exists significant doubt. This includes a mirror inscription, that quotes a line from the *Cang Jie pian*, but has altered the content to fit its own unique rhyme scheme. Another example is content from the sexagenary cycle, found in the SQZ cache, which Zhang Cunliang 張存良 argues belongs to the *Cang Jie pian*, but this is suspect.

Column EDITION

This column differentiates between various editions of CJP, as outlined above: Pre-VT for the edition prior to the village teachers' editing; VT for the version produced by the village teachers; and SQZ for the edition based on VT that also appends a three-character rhyming commentary. Mirror and 干支 are given for the mirror inscription and sexagenary cycle respectively.

Column STANZA

Chapters are taken as the "stanzas" for the *Cang Jie pian*. Each chapter participates in a single overarching rhyme scheme. Often times, however, these rhyme schemes continue across chapters as well. On this, please see the explanation for the "Rhyme ID Index" sheet under Column RHYMEIDS and note the correlations in chapters across editions on the "Pre-VT & VT Line Index" sheet. None of our manuscripts offer a complete text of the *Cang Jie pian*, making reconstruction necessary when determining units of textual division.

For the Pre-VT edition, our longest and most complete witness is the PKU ms. It explicitly titles a number of chapters, and at times also records character counts summarizing chapter lengths. The structure for the Pre-VT *Cang Jie pian* therefore is derived primarily from the PKU ms. When a chapter title is written on the PKU ms, this is used for the stanza name (e.g., 漢兼). When a partial title appears on the PKU ms, if the missing character can be supplemented by comparison to other sources, this is added inside rectangular brackets ([ ], e.g., [賞]祿); if the missing character cannot be supplemented, it is left blank with the character □ as a gap filler (e.g., □輪). When a title is not extant on the PKU ms, but is explicitly mentioned in other sources, this is also given within rectangular brackets ([ ], e.g., [爰歷], a title mentioned in the *Hanshu* 漢書 and *Shuowen jiezi* 說文解字). On a few occasions, a title both missing in the PKU ms and not mentioned in other sources may be suggested based on a comparison to the VT edition and our knowledge of the title conventions governing the PKU ms. This is

signaled by placing the title within rectangular brackets and adding an asterisk (*) beforehand (e.g., [*窒竅]).

There are nine chapters in the Pre-VT edition for which titles are missing completely. These are labelled “Pre-VT #1,” “Pre-VT #2,” etc. At times, content found on Pre-VT sources (namely the PKU and FY mss) cannot be located definitively within a known Pre-VT chapter. In such cases, the content is assigned to “Pre-VT Unknown” and then consecutively numbered. If we can hypothesize possible chapters to which the content may belong, these options are given in parenthesis: “Pre-VT Unknown #1 (爰歷, #4, or 機杼),” “Pre-VT Unknown #2 (爰歷, #4, or 機杼),” and “Pre-VT Unknown #3 (齎賻 or #5).” If not, it is left with just the number: “Pre-VT Unknown #4,” “Pre-VT Unknown #5,” etc.

For the VT edition, the *Hanshu* and *Shuowen jiezi* argue that village teachers divided the text into 55 chapters, each 60 characters in length. Based on this description, stanzas are labelled as VT 1, VT 2, VT 3… VT 55. Our longest and most complete witness is the HB ms. This manuscript consists of wooden boards, which each bear 60-characters of text written in three columns. Our assumption is that a single board corresponds to one chapter of the VT edition; this is supported by a comparison of the textual divisions across the HB board to our prior understanding of VT chapter divisions (see for instance JY 9.1, and the discussion in Foster forthcoming). On the top of the HB boards, a numerical label is written (e.g., 第一 or “1st”). This presumably numbers the VT chapter for the content on the board, and serves as our overall guide for placing content in a given chapter. Note however that these labels are often difficult or impossible to discern in the published photographs, and the proposed transcriptions given by Liu Huan 劉桓 can be erroneous. We judiciously adopt different arrangements for the VT edition, proposed by other scholars (e.g., the board Liu labels as 10 is treated as content for VT 20; the one Liu labels as 53乙 is treated as VT 55). With further research, our arrangement is liable to change.

Again there are times when certain content found on VT sources (namely the HB and SQZ mss) cannot be located definitively within a known VT chapter. In such cases, the content is assigned to “VT Unknown” and then consecutively numbered. If we can hypothesize possible chapters to which the content may belong, these options are given in parenthesis: “VT Unknown #1 (22, 28 or 32),” “Pre-VT Unknown #2 (22, 28 or 32),” etc. If not, it is left with just the number: “VT Unknown #4,” “VT Unknown #5,” etc. One unique situation requires explanation: the content on HB 42 likely precedes the content on HB 43甲, but we cannot determine if this pair corresponds to VT chapters 41+42 or VT chapters 43+44. We therefore title both chapters as #3, but further designate them as “a” versus “b” to communicate relative order: “VT Unknown #3a (41 or 43),” and “VT Unknown #3b (42 or 44).”

For the SQZ edition, because the base text follows the VT edition, chapter titling likewise will correspond to that of the VT edition: VT 1, VT 2, VT 3, etc. The only exceptions are when unknown content on the SQZ ms is too fragmentary to propose line breaks. In such cases, because it is uncertain if the content is from the VT base text or the SQZ commentary, the chapter is labelled "SQZ Unknown" and consecutively numbered: "SQZ Unknown #1," "SQZ Unknown #2," etc.

Column LINE_IN_SOURCE

This gives the text found on the manuscript cited in the corresponding SOURCE column. No paratextual features are included, such as chapter titles or character counts. Any punctuation on the manuscript is removed as well. Unless otherwise noted, the transcriptions given in "CJP Rhyming Data" follow those given in: Beijing daxue chutu wenxian yanjiusuo 2015 (for PKU), Hu and Han 1983 (for FY), Liu 2019 (for HB), Zhang 2016 (for JYX EPT 50.1), and Zhang 2015 (for SQZ). Future versions of "CJP Rhyming Data" will take into account all proposed transcriptions for each word, as debated in the scholarly literature. Our preference is to record the strict transcriptions, not interpretative, especially for the rhyme words. Strict transcriptions do not communicate when scribal errors, alternative forms, or loaning effect our reading. When this information potentially impacts the pronunciation of a word in a rhyming position, it is discussed in the "NOTES" column. For content outside of rhyming positions, no comment is given.

Occasionally the published transcription for a character is difficult or impossible to type. If the character falls outside a structural rhyme, we revert to the interpretative readings given in the publication's annotations. If no interpretative reading exists, or if it is important to retain the spelling of the strict transcription (especially for rhyme words), we describe the character with ⿰ symbols and regular *kaiti* 楷體 form components. Thus ⿰歹易 spells the rhyme word for the base text of VT 12 line 7 on SQZ C021, even though this is interpreted by Zhang Cunliang as 殤. This can be used to describe partial characters as well. For instance, FY C066 has ⿰□翏 as the partial remains of the rhyme word for Pre-VT ?#7, line n.

Column LINE_ORDER

As described in List et al. (2019), LINE_ORDER is "A numerical value that provides the order of the lines of a poem in a given stanza." For "CJP Rhyming Data," numerical values designate absolute line number: 1 is the first line of the chapter, 2 is the second, and so forth. When only relative line order can be determined, this is represented by consecutive letters in the alphabet: a, b, c, etc. Consider for instance PKU 45. We may

speculate that it belongs to the chapter Pre-VT #6. A character count of 144 is found after this final line on PKU 45, which moreover tells us that it was the thirty-sixth line of the chapter. Thus the LINE_ORDER value is 36. With PKU 14, while it is likely that the content on this strip belongs to Pre-VT #8, we do not know where precisely it fits within this chapter. The five lines written on PKU 14 are therefore labelled a-e for LINE_ORDER, as only their relative order is secure.

We can exploit LINE_ORDER to document variants found on other manuscripts (List et al. 2019, p. 39). This is accomplished by designating the same LINE_ORDER to multiple entries under the same STANZA, with each given a different SOURCE. For example, the database gives five different sources for the rhyme word on VT 1 line 2, showing 嗣, 子 and 生 as variants. This allows for direct comparison of variants, but only from the same edition (e.g., Pre-VT versus VT and SQZ). Often it is important to compare parallel text across editions. For example, Pre-VT 顓頊 line 4 is only attested on PKU 46, where the rhyme word is 襄; but there is parallel text for this line in VT 11 line 4, found on HB 11乙, which gives the rhyme word as 鑲. To help preserve these relationships, two approaches have been adopted. First, “CJP Rhyming Data” lists parallel or connected content close to one another on the sheet, allowing for easy visual reference and grouping together IDs for like material. “Pre-VT & VT Line Index” directs the user to where a given Pre-VT or VT chapter begins and ends relative to the other edition, offering a rough guide for where to consult when looking for content across editions. Second, the numbering of RHYMEID also suggests content relationships. We are currently discussing how to handle this more systematically in future versions.

Column RHYMEIDS

Note that RHYMEIDS is different from ID. RHYMEIDS describes where in a given line a rhyme word is present. For each character position in the line, a numerical digit is assigned. When the word is not part of a rhyming relationship, it is assigned a 0. For example, the RHYMEIDS for the four characters in VT 1 line 1, 蒼頡作書, is 0 0 0 0, telling us that none of the words are rhyming. When the word is part of a rhyming relationship, it is assigned a number (1, 2, 3…), with each word participating in that rhyme scheme sharing in the same number. Thus VT 1 line 2, 以教後嗣, has 0 0 0 1 for the base text, which describes the fourth word, 嗣, as rhyming. Compare this to VT 1 line 4, 謹慎敬戒, which also has 0 0 0 1. This means that the fourth word of this line, 戒, is a rhyme word. Furthermore, it participates in the same rhyme scheme as 嗣, since both have the RHYMEID of 1. If it is uncertain which words participate in a rhyme scheme, we write a question mark instead of a numeral. Take for instance VT Unknown

#31, line b, 被衾襖綒, found on SQZ C104. Because we do not know if 綒 is a rhyme word, the RHYMEID for the line appears as: 0 0 0 ?.

Because the Pre-VT, VT, and SQZ editions of the Cang Jie pian derive from similar content and include parallel text, often rhyme schemes are repeated or extended through comparison across different editions. We have attempted to preserve this information indirectly in assigning RHYMEIDs. VT edition rhyme schemes run from 1-55 (representing each of its fifty-five chapters). For VT content that rhymes, but for which we do not know the corresponding chapter, RHYMEIDs then count up from 56.

The Pre-VT edition of the Cang Jiepian has longer chapters, which often incorporate multiple VT chapter rhymes schemes per single Pre-VT chapter. For example, the content of the Pre-VT 顓頊 chapter parallels that of VT 11-13. To help communicate this association, Pre-VT RHYMEIDs add a 0 to the RHYMEID for that of the first corresponding VT chapter. Because the Pre-VT 顓頊 chapter begins with parallel content to VT 11, its RHYMEID is 110. Note that a 0 is added in front of Pre-VT RHYMEIDs which parallel VT 1-9 content: e.g., Pre-VT [賞]祿 has 030, because it begins with content seen on VT 3 as well. For Pre-VT content that rhymes, but for which we do not know the corresponding chapter, Rhyme IDs then count up from 1000. Note that the relationship between VT Rhyme ID 10 and Pre-VT Rhyme ID 100 is unique, in that scholars have proposed different line breaks for the parallel content found here; this means that these two sections are textually related but potentially offer conflicting rhyme schemes. This is a problematic section of the *Cang Jie pian* which we will treat in more depth in a future article.

The SQZ ms is based on the VT edition, but adds three-character rhyming commentary to the four-character base text. This creates two connected yet separate rhymes schemes, given as R:1 and R:2, for the base text and commentary respectively. Since the SQZ edition is based on the VT edition, the R:1 RHYMEIDs correspond to those of the given VT chapter. The R:2 RHYMEIDs, however, add a 1 to the end of that same VT edition RHYMEID. For example, the strip SQZ C052 bears content related to VT 20 line 5, but with additional commentary: 偃黿運糧 (base text) + 載穀行 (commentary). Its RHYMEID therefore is: 0 0 0 20 0 0 201. This tells us that the fourth character, 糧, is the first rhyme word (R:1) and that it participates in the rhyme scheme for VT 20; the seventh character, 行, is the second rhyme word (R:2), and participates in the rhyme scheme for the SQZ commentary to VT 20. Note that again a 0 is added in front of SQZ R:2 RHYMEIDs based on VT 1-9 content: e.g., SQZ commentary to VT 1 is 011. For SQZ content that rhymes, but for which we do not know the corresponding VT chapter, Rhyme IDs then count up from 2000.

Finally, the Han mirror inscription quotes a line from the *Cang Jie pian*, but alters the rhyme word to fit a rhyme scheme unique to its own content. Its RHYMEID is assigned to 9999.

Column ALIGNMENT

For lines with a rhyme present, the Chinese character writing the rhymed word is replaced with a reconstruction of that word's pronunciation. This reconstruction follows William Baxter and Laurent Sagart's 2016 Old Chinese (https://ocbaxtersagart.lsait.lsa.umich.edu). When multiple reconstructions are possible, each is recorded in NOTES. In such cases, the pronunciation given in ALIGNMENT is a preliminary judgment about the word intended in the linguistic context of the line (including the possibility of loaning), but this can be ambiguous and awaits final analysis. It should not be taken as our definitive statement on the text's meaning.

When a reconstruction is not available for the rhyme word in Baxter and Sagart, an alternative reconstruction is found. As an expedient, this often entails substituting in a reconstruction given by Baxter and Sagart for a word that is both from the same *xiesheng* 諧聲 series and has the same Middle Chinese pronunciation. If such a substitute is not available, then the reconstruction is for the character's phonetic component or based on some other phonetic information. This is of course methodologically problematic, and only serves as a placeholder until further analysis. If there is no or only partial evidence for the rhyme word, making a reconstruction impossible, a question mark is written instead.

Columns R:1 and R:2

These columns provide a convenient way to identity the rhyme schemes present in the *Cang Jie pian* manuscripts. R:1 is based on the formulaic rhyming in CJP; R:2 is based on the formulaic rhyming in CJP SQZ. Internal rhymes or other rhyme schemes can be added in future versions with the addition of new columns (R:3, R:4, etc.). If no evidence for a word exists in a known rhyming position on our manuscript source, then we write a ? in the column. Partial evidence is documented by spelling out the orthography via □ symbols.

Column SOURCE

When dealing with multiple manuscript witnesses, it is necessary to reference specific sources for the content. Labels are provided for individual strips bearing representative text (recorded in LINE_IN_SOURCE). A key for these labels may be found at the top of the "Bibliography" sheet. If multiple strips are re-pieced together as testimony to a

single line, this is documented in SOURCE with a + sign. For example, the SOURCE for SQZ VT 1, line 5, is SQZ C003 + SQZ C004, since the final two characters of the commentary for this line are missing on SQZ C003, but supplemented by SQZ C004.

"CJP Rhyming Data" incorporates PKU, FY, HB, JYX EPT50.1, and SQZ in full. When variants for rhyme words appear on other sources, we add entries for them in the database. Other witnesses that write rhyme words, but do not offer a variant, are documented in the NOTES column, but not given an unique entry. We do not document strips and fragments from other sources if they lack information about rhyme words; eventually we hope to include all *Cang Jie pian* manuscript finds, but this must await a future version. Currently (October 2020), only a brief report is available for the 2018 discovery of *Cang Jie pian* material at the Chengba 城壩 site in Quxian 渠縣, Sichuan (Sichuan sheng wenwu kaogu yanjiuyuan et al. 2019). We will include this find in "CJP Rhyming Data" once the data is published in full, should it bear new rhyming data.

It must be emphasized that the PKU and HB witnesses are purchased manuscripts. They were not secured through scientific archaeological excavation, and lack proper provenience. Foster 2017b argues for the authenticity of the PKU ms. In the fall of 2019, it was announced that a private collector possessed another *Cang Jie pian* manuscript, written across numbered wooden boards, labelled as the HB witness in "CJP Rhyming Data" (Liu 2019). HB has not yet been properly authenticated, although a number of scholars assert that it is indeed a genuine artifact (Fukuda 2020, Zhang 2019, etc.). The data for HB was published in Liu 2019, but unfortunately Zhonghua shuju 中華書局 has recalled the book and it is no longer available for purchase. We have decided to include the HB ms in "CJP Rhyming Data" in part to make transcriptions available for interested scholars. Due caution is still warranted, however, when using this source for our research.

Column SOURCE_COLLECTION

Finally, we have added in another column called SOURCE_COLLECTION to enable users to filter between different manuscript sources (FY, PKU, HB, SQZ, JYX, etc.). In this way, a user may see the entire content from a single manuscript. This is necessary especially for multi-piece manuscripts, where the documentation in SOURCE alone would give the user only one piece of a larger manuscript. EDITION is likewise unsuitable as filter, as it usually will incorporate multiple manuscript witnesses.

Summary

"CJP Rhyming Data" is an initial attempt to test our standardized rhyming framework on a larger and more complex corpus of manuscript data. Already, in the input of this data, certain modifications have been made, anticipating users' needs. The inclusion of

EDITION, SOURCE and SOURCE_COLLECTION as columns, and the sheets “Rhyme ID Index” and “Pre-VT & VT Line Index,” are examples, where we attempt to preserve textual and material relationships in the dataset that are difficult to express in the prior framework. Inevitably, with “CJP Rhyming Data” now compiled, other unforeseen needs and missing functionality will be laid bare. To this end, we treat this study as an ongoing collaborative project and anticipate frequent updates to the dataset. Documenting these changes, and their reasons, will itself continuously demonstrate the various strengths and weakness to our framework.

Dataset

The dataset itself is curated on GitHub, where it can be found at https://github.com/digling/cjp-data/, and has been released in a first version to Zenodo, where it can be found at https://doi.org/10.5281/zenodo.4084859. I thank Johann-Mattis List for his help with managing this dataset.

References

Baxter, William H., and Laurent Sagart. *Old Chinese: A New Reconstruction*. New York: Oxford University Press, 2016.

Beijing daxue chutu wenxian yanjiusuo 北京大學出土文獻研究所, ed. *Beijing daxue cang Xi Han zhushu* (*yi*) 北京大學藏西漢竹書(壹). Shanghai: Shanghai guji chubanshe, 2015. (PKU).

Foster, Christopher J. “Study of the Cang Jie pian: Past and Present.” PhD Dissertation, Harvard University, 2017.

Foster, Christopher J. “Introduction to the Peking University Han Bamboo Strips: On the Authentication and Study of Purchased Manuscripts.” *Early China* 40 (2017): 167-239.

Foster, Christopher J. “The Shape of the Text: Gu Prisms and Han Primers.” In Susan Blader, Constance Cook, and Christopher J. Foster, eds. *Thinking About Early China with Sarah Allan*, forthcoming.

Fukuda Tetsuyuki. *Setsubun izen shogakusho no kenkyo* 說文以前小学書の研究. Tokyo: Sobunsha, 2004.

Hu Pingsheng 胡平生 and Han Ziqiang 韓自強. “Cang Jie pian de chubu yanjiu 蒼頡篇的初步研究.” *Wenwu* 文物 2 (1983): 35-40. (FY).

Jiandu zhengli xiaozu 簡牘整理小組, ed. *Juyan Han jian* 居延漢簡. 4 vols. Taipei: Zhongying yanjiuyuan lishi yanjiusuo, 2014-17. (JY)

Liang Jing 梁靜. *Chutu Cangjie pian yanjiu* 出土蒼頡篇研究. Beijing Kexue, 2015.

List, Johann-Mattis, Nathan. W. Hill, and Christopher J. Foster. “Towards a Standardized Annotation of Rhyme Judgments in Chinese Historical Phonology (and Beyond).” *Journal of Language Relationship* 17 (2019): 26-43, https://doi.org/10.31826/jlr-2019-171-207.

Liu Huan 劉桓. *Xinjian Han du Cang Jie pian Shi pian jiaoshi* 新簡漢牘蒼頡篇史篇校釋. Beijing: Zhonghua shuju, 2019.

Schuessler, Axel. *Minimal Old Chinese and Later Han Chinese: A Companion to Grammata Serica Recensa*. ABC Chinese Dictionary. Honolulu: University of Hawai’i Press, 2009.

Sichuan sheng wenwu kaogu yanjiuyuan 四川省文物考古研究院 and Quxian lishi bowuguan 渠縣歷史博物館, ed. “Sichuan Quxian Chengba yizhi 四川渠縣城壩遺址.” *Kaogu* 考古 7 (2019): 60-76.

Tsien, Tsuen-hsuin. *Written on Bamboo & Silk: The Beginnings of Chinese Books & Inscriptions*. 2nd ed., with afterword by Edward Shaughnessy. Chicago: University of Chicago, 2013.

Zhang Chuanguan 張傳官. “Tantan xinjian mudu Cang Jie pian de xueshu jiazhi 談談新見漢牘蒼頡篇的學術價值.” Fudan University Center for Research on Chinese Excavated Classics and Paleography 復旦大學出土文獻與古文字研究中心 Article Database. 25 December 2019, http://www.gwz.fudan.edu.cn/Web/Show/4510.

Zhang Cunliang 張存良. “Shuiquanzi Han jian Cang Jie pian zhengli yu yanjiu 水泉子漢簡蒼頡篇整理與研究.” PhD Dissertation, Lanzhou University, 2015. (SQZ)

Zhang Defang 張德芳, et al. *Juyan xin jian jishi* 居延新簡集釋. Lanzhou: Gansu wenhua chubanshe, 2016, 7 vols. (JYX).

Zhongguo jiandu jicheng bianji weiyuanhui 中國簡牘集成編輯委員會, ed. *Zhongguo jiandu jicheng* 中國簡牘集成. Lanzhou: Dunhuang wenyi chubanshe, 2001+, vol.14 & 18. (FY).

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Towards a refined wordlist of German in the Intercontinental Dictionary Series

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

The post introduces the plan to create a new wordlist for the Intercontinental Dictionary Series and starts from creating a new concept list that refines underspecified entries in the IDS along with a refined wordlist of German that offers data in IPA transcriptions.

For a long time, I have been wondering about the origin of the German wordlist in the Intercontinental Dictionary Series (Key and Comrie 2016). Not only are many of the words given as translations for the large concept list of 1310 items very archaic variants, which are no longer in use, we also find many annoying problems, such as unusual spellings (consequently avoiding the letter "ß", which is still in use, even if some people think differently), wrong translations, and, of course, no phonetic transcriptions. Already during my doctoral studies, I therefore started to work on a refined list, but I soon had so many other things on my plate, that I never really managed to finish this work. Recently, however, I realized that my previous work which I had done years ago was far more complete than I had thought, and I had even added information on potential borrowings, extracted from Kluge's (2002) etymological dictionary. Given that this list can come in handy in various ways, I decided to finish the work and publish the list officially in a very first version.

I should add that the work cannot be considered complete, since I realized that many duplicate items have not yet been thoroughly checked, and some of the archaic terms which are no longer in use are still in the list (e.g., Oheim, which is an archaic term for uncle in German). Furthermore, there may be certain problems in the phonetic transcriptions, which I by then extracted from the CELEX database (Baayen et al. (1995)), and which contain several problems, which I checked manually, but there is no guarantee I succeeded completely, although I made sure all transcribed items conform to the B(road)IPA standard proposed by the CLTS project (Anderson et al. 2018).

Computer-Assisted Language Comparison in Practice

Volume 3, Number 10
https://calc.hypotheses.org/2545

While reviewing my transcriptions, which I also expanded by adding morpheme boundaries, I further realized that the IDS glosses have so far only been translated to Spanish, Russian, and Portuguese, with no German translation of the elicitation glosses being available so far. Therefore, I decided to add these translations as well. Again, it is not clear whether I did a completely satisfying job here, but it is probably enough to be published in the form of a first version.

While I first planned to follow the Concepticon mappings as they are given for the IDS list in the Concepticon project (List et al. 2020), I realized — what people have realized before — that the glosses in the IDS are often unfortunate, since they are so broad that people feel forced to offer many different translations, often including both nouns and verbs. In order to allow for a more consistent mapping to the Concepticon project, I therefore modified these entries, keeping the original keys from the IDS data, and adding more specific elicitation glosses in German as well as the corresponding Concepticon identifiers. In some cases, I found concepts which are not yet available in the Concepticon but should ideally be added in the future. These cases were marked with an asterisk in the concept list.

All in all, this dataset thus comes in two flavors. There is a concept list with German elicitation glosses for almost all of the concepts we find in the IDS and a large amount of the concepts we find in the WOLD project (Haspelmath and Tadmor 2009), and there is a *wordlist*, which provides translation equivalents for these concepts and offers phonetic transcriptions. The dataset itself is available in the form of a GitHub Gist, which you can find at https://gist.github.com/LinguList/cfa4ab9b2b168fbc07d8247352fb6039.

References

Anderson, Cormac and Tresoldi, Tiago and Chacon, Thiago Costa and Fehn, Anne-Maria and Walworth, Mary and Forkel, Robert and List, Johann-Mattis (2018): A Cross-Linguistic Database of Phonetic Transcription Systems. Yearbook of the Poznań Linguistic Meeting 4.1. 21-53.

Baayen, R. H. and Piepenbrock, R. and Gulikers, L. (eds.) (1995): The CELEX Lexical Database. Version 2. Philadelphia:

Johann Mattis List and Christoph Rzymski and Simon Greenhill and Nathanael Schweikhard and Kristina Pianykh and Annika Tjuka and Mei-Shin Wu and Robert Forkel (2020): Concepticon. A resource for the linking of concept lists (Version 2.3.0). Version 2.3.0. Max Planck Institute for the Science of Human History. Jena: https://concepticon.clld.org/.

Haspelmath, Martin and Tadmor, Uri (2009): Loanwords in the worldś languages. Berlin and New York:de Gruyter.

Key, Mary Ritchie and Comrie, Bernard (2016): The intercontinental dictionary series. Leipzig:Max Planck Institute for Evolutionary Anthropology.

Kluge, Friedrich (2002): Etymologisches Worterbuch der deutschen Sprache . Berlin:de Gruyter.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Computing colexification statistics for individual languages in CLICS

Tiago Tresoldi
Department of Linguistic and Cultural Evolution
Max Planck Institute for the Science of Human History

The post presents new statistics on the CLICS database which were computed upon request by colleagues and published along with this blog post.

In the last two weeks we had a renewed interest in colexifications, especially in the third generation of the "Database of Cross-Linguistic Colexifications" (Rzymski, Tresoldi, et al., 2020). The attention was due to two different and independent requests in few days. For those unfamiliar, the concept of "colexification" (François, 2008) refers to instances in which a language uses the same lexeme to express more than one comparable concept (e.g., Russian дéрево, which can mean both "tree" and "wood"). The CLICS project, first developed by List et al. (2014), is an offspring of the transparent approaches to standardization, aggregation, and curation of linguistic data that have been promoted within the CLDF framework (Forkel et al., 2018). It uses standardized lexical databases to identify "colexification networks".

In the first request, Ezequiel Koile, who was studying geographic linguistic patterns, asked us if it was possible to collect a list of which languages colexify specific pairs of concepts. In addition to this data not being aggregated directly in our software library, despite being indirectly accessible via the web interface, his request involved concept pairs that are not necessarily included in the database. Designed to identify communities of cross-linguistic patterns, the algorithm for CLICS excludes concept pairs with weaker signals, such as those due to pure homophony or to problems in data, as well as those that don't have enough statistical significance to be part of a community.

Computer-Assisted Language Comparison in Practice

Volume 3, Number 11
https://calc.hypotheses.org/2552

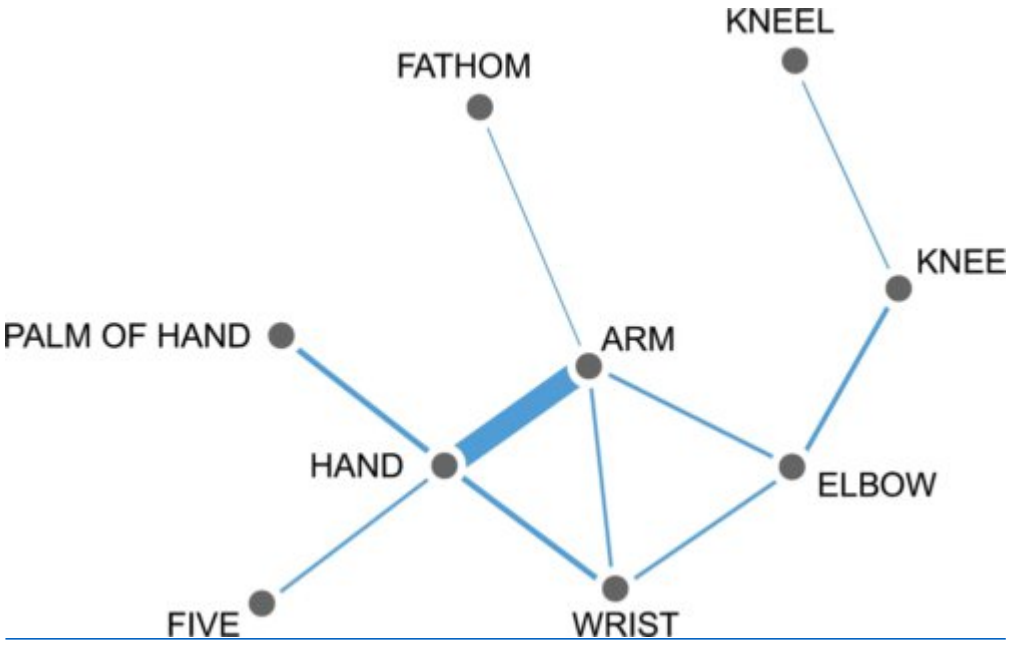

Figure 1: A colexification network (from Rzymski, Tresoldi, et al. 2020)

Days later, another colleague, David Gil, asked if we had ever collected data on the individual colexification frequency per language. He was looking for a continuous typological variable of "tendency for a language to have colexifications". I knew that this information was available internally, but it was not offered explicitly to the public, nor there was a straightforward way to retrieve it from the existing output or from the web interface.

CONCEPT_A	CONCEPT_B	NUM_LANGUAGES	LANGUAGES
A LITTLE [2924]	SMALL [1246]	1	Hahl Mongolian [halh1238]
A LITTLE [2924]	WHAT [1236]	1	Dutch [dutc1256]
ABANDON [1097]	THROW AWAY [3696]	2	Ere (China) [eree1240]; Gdongbrgyad-Kamnyu [gdon1234]
[...]	[...]	[...]	[...]
TREE [906]	WOOD [1803]	300	Abui [abui1241]; Aché [ache1246]; Adang [adan1251]; Adonara [adon1237]; Adyghe [adyg1241]; ... Zacatepec Chatino [zaca1242]; Zaiwa [zaiw1241]; Zuojiang Zhuang [zuoj1238]
[...]	[...]	[...]	[...]
YOUNGER SIBLING [427]	YOUNGER SISTER (OF WOMAN) [2421]	13	Biak [biak1248]; Busoa [buso1238]; Dadua [dadu1237]; ... Ringgou [ring1244]; Sika [sika1262]
YOUNGER SIBLING [427]	YOUNGER SISTER [1761]	7	Central Kanuri [cent2050]; Dutch [dutc1256]; Gurindji [guri1247]; Plateau Malagasy [plat1254]; Swahili [swah1253]; Takia [taki1248]; Yaqui [yaqu1251]
YOUR (PLURAL) [2274]	YOUR (SINGULAR) [732]	13	Apali [apal1256]; Asas [asas1240]; Korak [kora1296]; Kulsab [fait1240]; ... South Adelbert [sout3148]; Utarmbung [utar1238]

Table 1: Selection of languages and information from the "languages per colexification" results.

As we organize the CLICS data in a relational manner and as the algorithm follows a well-structured sequence of steps, using established methods for detecting communities, it was easy to address both requests. The first one was a matter of collecting the languages for each colexification before the community detection. The result is a single table, with concepts listed alphabetically (so we know to search for “TREE and WOOD” and not “WOOD and TREE”) and languages equally ordered. The most common colexifications are confirmed, but the results include a “long tail” of concept pairs with only a single colexification. Information from other resources, such as CLICS itself or Glottolog, can be easily aggregated thanks to the reference catalogs from the CLDF ecosystem.

The second request involved a little more effort in coding. Not only the amount of data for each language varies (and, even more, varies for the same language in different dataset), but also the mutual coverage when considering all varieties is very small. To discuss which languages have most colexifications implies knowing the potential number of them, and there is no point in counting a concept pair if we data for that pair is not available. Likewise, it is necessary to consider whether we should count over all possible colexifications (as in the case above, more concerned with areal or family tendencies) or just those that are over the thresholds to be included in CLICS. As expected, the ratios of observed colexifications are overall very low, in the order of fractions of percentage points. It was not enough to collect these values, we also needed to offer them in a usable way, properly normalized. Ratios were thus adjusted, offering a convenient number between zero and one for each language, with higher values expressing a greater tendency to colexify.

LANG_KEY	GLOTTOCODE	NAME	ADJUSTED_RATIO_ALL	ADJUSTED_RATIO_THRESHOLD
tng-abaga	abag1245	Abaga	0.0656	0.0000
huber-Achagua	acha1250	Achagua	0.0995	0.2672
ids-219	basq1248	Basque	0.1144	0.3392
diacl-43200	mode1248	Modern Greek	0.0802	0.2244
tls-Kipogoro	pogo1243	Pogolo	0.1148	0.3347
ids-414	zuoj1238	Zuojiang Zhuang	0.1775	0.3610

Table 2: Selection of languages and adjusted ratios from the “language colexification affinity” results.

With these data, it was possible to rely on the geographic coordinates provided by Glottolog and plot a map with the adjusted ratio for each language. As always, a visual exploration suggests patterns and trends that can later be statistically tested. This is the case here, with some areal patterns clearly visible. Polynesian languages seem to be among those most prone to colexification, with high ratio founds also in Southeast Asia and the Amazon (despite some outliers). The ratios for Indo-European languages suggest an intermediate position, and the languages of Papua, regardless of family, appear to be among those with the lowest tendency in the world.

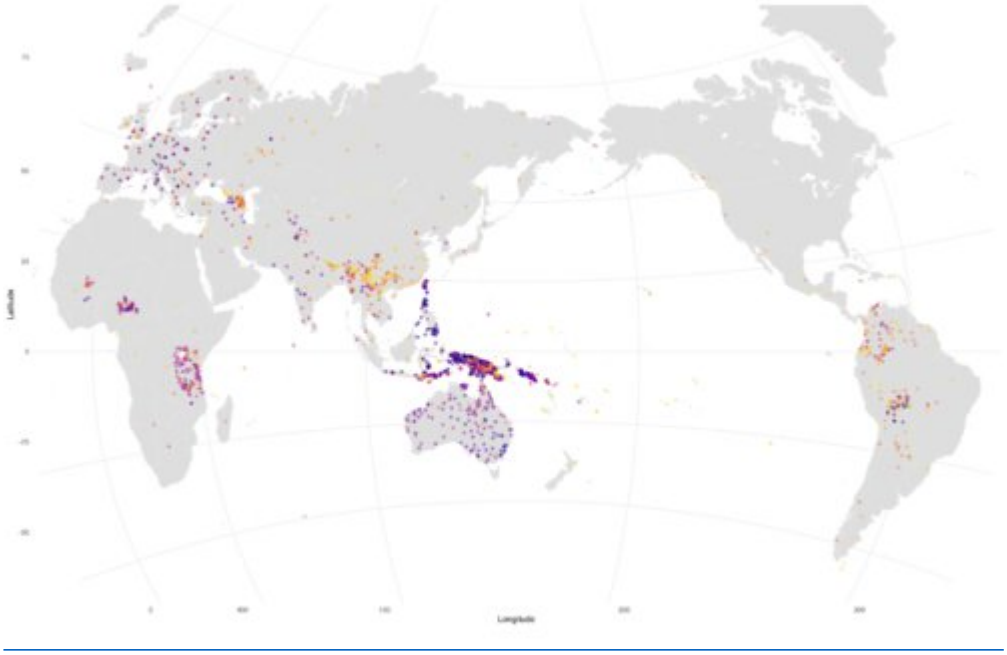

Figure2: World map of language colexification affinity

It's never superfluous to remember that these data demand the same care of any linguistic research: some characteristics, especially for languages for which we have a single source, may be due to biases in their collection, or even problems in data management. Nonetheless, it is satisfying to watch science at work. First, we have made the efforts of hundreds of linguists over the centuries accessible by good data practices. The results of such practices allowed to publish a resource like CLICS, which other researchers have identified as a viable answer for their questions. In collaboration, we could offer the data they needed, which might nurture the cycle again. But there is a significant difference because of the "data FAIRness" (Wilkinson et al., 2016) approach we have adopted: the sources and the manipulation are transparent and reproducible, so that solutions can be improved and errors can be corrected as a normal part of the research process.

The code for collecting these data has already been merged to pyclics, and can be used from the command-line "`clics`" tool. The two tables here discussed, along with the vector maps for both adjusted ratios, are available on Zenodo (DOI: 10.5281/zenodo.4148257).

References

Forkel, R., J.-M. List, S. Greenhill, C. Rzymski, S. Bank, M. Cysouw, H. Hammarstrom, M. Haspelmath, G. Kaiping, and R. Gray (2018): Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. Scientific Data 5.180205. 1-10. DOI: 10.1038/sdata.2018.205.

François, A. (2008): Semantic maps and the typology of colexification: intertwining polysemous networks across languages. In Vanhove, M. (ed.) From polysemy to semantic change. Benjamins: Amsterdam, 163–215.

List, J.-M., Mayer, T., Terhalle, A. & Urban, M. (2014): CLICS database of crosslinguistic colexifications. Forschungszentrum Deutscher Sprachatlas: Marburg, DOI: 10.5281/zenodo.1194088.

Rzymski, C., T. Tresoldi, S. Greenhill, M. Wu, N. Schweikhard, M. Koptjevskaja-Tamm, V. Gast, T. Bodt, A. Hantgan, G. Kaiping, S. Chang, Y. Lai, N. Morozova, H. Arjava, N. Hubler, E. Koile, S. Pepper, M. Proos, B. Epps, I. Blanco, C. Hundt, S. Monakhov, K. Pianykh, S. Ramesh, R. Gray, R. Forkel, and J.-M. List (2020): The Database of Cross-Linguistic Colexifications, reproducible analysis of cross- linguistic polysemies. Scientific Data 7.13. 1-12. DOI: 10.1038/s41597-019-0341-x.

Wilkinson M. D., M. Dumontier, IJ. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. S. Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B, Mons -Show (2016): The FAIR Guiding Principle for scientific data management and stewardship. Scientific Data 3.160018. DOI: 10.1038/sdata.2016.18.

Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Possibilities of Digital Communication in Linguistics (How to do X in Linguistics 2)

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

I noticed that scientists deal with digital communication very differently: some avoid all sorts of platforms, others are much more present and involved in the discussions that are taking place in the online world. Digital communication as a linguist (or scientist in general) also includes sharing your research output. This does not have to be an article in a high-ranking journal. As a student, you can start by publishing your thesis, conference presentation slides, or a preprint. In this post, I'll illustrate some of the possibilities that linguists and other researchers have to discuss and share their work.

Why is it Important to be Visible Online?

Research needs to be shared so that your colleagues can find, read, and discuss your contributions to the field. Especially, if you are a doctoral student or a junior researcher! When you are a student, often only your supervisor knows about the amazing work you do. In contrast, senior scientists benefit from their extensive network. Newcomers can therefore use well-established platforms to expand the readership of their work.

Where to Start?

Start simple: most universities or institutes provide the opportunity for setting up a personal profile on the university/institute website. Fill in all the blanks and add the link of your personal page to your e-mail signature. But be aware that the site may disappear when you change to another university/institute, so make sure you save your data before you leave.

If you feel up for a challenge, you can also create your own personal website. There are various platforms that offer templates to set up and host a personalized website (e.g.,

Computer-Assisted Language Comparison in Practice

Volume 3, Number 12
https://calc.hypotheses.org/2556

GoDaddy). A completely free option is GitHub pages which provides templates and a vast amount of freely available inspirations. The page is built from a GitHub repository that is rendered into a website with the suffix github.io. To see how it looks like, you can check out my repository and website. No matter which solution you choose, make sure that you are able to update the content of your website by yourself. It's no use having the most creative website if you have to ask someone every time you want to correct a typo or add an article. Some well-established researchers get along quite well without a personal website. So you may only use your university/institute website or one of the platforms, I'll discuss below.

Online Platforms (for all Scientists)

Many platforms offer researchers the opportunity to share their work. However, some of them are designed like social media platforms (e.g., ResearchGate). That way one might run the risk of spending too much time and energy *following*, *updating*, and *posting*. Nevertheless, the platforms are useful to inform your colleagues about your newest presentation, preprint, or article.

- ResearchGate: It's free and easy to edit. In addition, you can connect, follow, and share your research with your colleagues. If you don't have an article yet, you can add the slides of your conference presentations or thesis.
- Academia: There is a free version, but the interesting statistics are hidden behind a paywall. I found that some of my colleagues use Academia instead of ResearchGate. That's why I joined both platforms.
- ORCID: This is basically your unique identifier as a researcher. Most journals will ask you to provide it when your article is accepted for publication.
- GoogleScholar: As soon as you publish your first article, you can set up your account.
- Twitter: It is very likely that you are already "on" Twitter even if you have no account (try googling "`YOUR NAME + Twitter`"). For example, when someone posts a picture of you presenting a talk. Twitter can be used as a social networking platform but I realized that many researchers also promote their most recent articles in a post. That way, you get informed about the newest literature in your field.

In general, it is essential to neatly organize your articles and share your work. You do not need to have an account for all these platforms and you should think about how much time you want to spend maintaining your online profiles. It would also be worthwhile to have one platform/website that you keep updated. You should also consider how you

want to share your work. If your article is open access, you can post the link instead of a PDF. So your colleagues can download the article directly from the journal webpage.

Online Platforms (for Linguists)

The following list of online platforms is of special interest for linguists and other scientists working in humanities or life sciences.

- The LINGUIST List: The website provides information about conferences, summer/winter schools, study programs, and mailing lists, specifically for linguists. The mailing lists are a great way to get in contact with your peers, ask them questions, and share your knowledge on incoming queries. In addition, LINGUIST List has a "Ask a linguist" section where you can ask or answer questions.
- LingBuzz: It can be used for a literature search of articles in linguistics and you can upload your own article for free. At the same time, it is a community space for linguists.
- Humanities Commons: A network platform for everyone working in humanities and a place to upload your thesis, working papers, presentation slides, and teaching material. Hcommons also offers the possibility to create a WordPress Web site.
- PsyArXiv: For researchers working in psycholinguistics or related fields, this service can be used to upload your preprint. The archive is supported by the Open Science Framework.
- Especially in the humanities, it is not yet common to publish preprints of your articles or preregister your hypothesis (see the website of the Center for Open Science). But good scientific practice should include making your research available to everyone.

More Ideas

There are also other ways to make people aware of your work. For example, you can record your next conference presentation and share it on platforms such as Vimeo or YouTube. It could be a slide show with narration in PowerPoint or a short video clip created with iMovie (for an example click here). If you don't have any upcoming conferences, you can start with a very short video in which you introduce your research profile (mine looks like this) or make an elevator pitch about your project.

An idea for which I received a lot of positive feedback is a virtual bookshelf. This can be easily integrated into your personal website and it sparks interesting conversations. There is a general interest in what other people read and often people pause to check out a friend/colleague's bookshelf. In addition, maybe a few of your students will get inspired to read some of your favorite books.

Keep in mind that developing an online presence takes time and is an ongoing process. Start with one thing. I'd suggest having one well-maintained profile rather than many half-finished ones.

Further readings

- Visibility: Build your online presence: Scholarly publishing
- 5 Methods to Develop Your Online Presence (for Researchers)
- How to build your online researcher identity and increase your impact