Using EDICTOR 2.0 to Annotate Language-Internal Cognates in a German Wordlist

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for Evolutionary Anthropology

With the recent publication of the new version of the EDICTOR application for the curation and creation of etymological dictionaries, several new features were introduced which target specifically the annotation of language-internal word families opposed to cross-linguistic cognates. While working on the EDICTOR update, I carried out intensive tests of the new features by annotating a German wordlist for language-internal cognates. In this post, I will quickly discuss some of the new features in EDICTOR 2.0 by showing some examples of the freshly annotated wordlist for German.

EDICTOR 2.0

After a longer time of silent development in which no official versions of the EDICTOR were published, I decided that it is time to assemble recent modifications and publish a new official version of the tool. This version is now available at the official EDICTOR website at https://digling.org/edictor/. For those interested in future versions of the EDICTOR, I recommend to use the development version, which you can find at https://lingulist.de/edictor.

Major features of the new EDICTOR version include:

1. a new panel allowing for the quick annotation of cognate morphemes by using cognate identifiers and morpheme glosses as well as the manual morpheme segmentation of words.
2. the indication of language subgroups, provided the data contains a column called “SUBGROUP” in the wordlist panel and in some of the alignment panels.
3. a new markup for cognate sets (mainly partial cognates), indicating whether they span a) several concepts, or are b) singletons, occurring only one time in the data.
4. the automated update of the data when working on multiple open instances for the same dataset with the help of the database version of the EDICTOR tool.

In the following, I will focus only on those features which are relevant for the annotation of language-internal cognates and word families.

**Horizontal and Vertical Language Comparison**

I have pointed out earlier how important it is to not forget about the importance of comparing language vertically when working on comparative wordlists (List 2018). Vertical language comparison is hereby understood as the comparison of the inherited structures within one and the same language, as opposed to the comparison of common structures across many different languages (horizontal language comparison).

When annotating comparative wordlists, we will often find situations in which a word is not only cognate with words in other languages, but also with parts of other words in the same language. If you have both “to obtain” and “to come” in your elicitation glosses, you cannot ignore that the German translations be-komm-en and komm-en share the lexical root komm. To handle this case of partial cognacy, you would have to indicate the separation of the data into morphemes by adding plus-markers (+) to your segmental representation (column TOKENS) in your wordlist, and then give the roots and the infinitive ending -en the same cognate identifier. If you want to be really clear, you should also use morpheme glosses (Hill and List 2017 and Schweikhard and List 2020) to indicate the meaning of the components, as shown in the following screenshot.

![Table showing German words derived from komm-](image)

**Figure 1:** German words derived from komm-

If you have both “language” and “to speak” as elicitation glosses in your comparative wordlist, you can also hardly ignore that the German translations Sprache and sprechen are also cognate. They are not strictly cognate, as we have outlined before (Schweikhard and List 2020), since they reflect non-concatenative morphology, due to the vowel alternation, but this does not mean that they are not cognates, and it would be a pity, if we did not find a way to indicate this regularly in our comparative wordlists. In order to
annotate this example, you could follow the same principle of segmenting the words and then assigning them to partial cognate sets along with morpheme glosses, as shown in the following figure.

![Figure 2: German words derived from sprech-](image)

Note one further modification, which we use here, to make the cognacy of [ʃ p r ɛ ç] and [ʃ p r a x] more explicit: we use the slash-construction by which we can indicate a source and a target phoneme in EDICTOR (and CLDF) to handle morphological alternations. This feature has been tested for some time now and allows us to handle various situations in which we want the computer to read one bit of information, while we want the humans to be able to read another bit. So instead of [ʃ p r a x] I write [ʃ p r a/ɛ x/ç]. This means that EDICTOR will internally treat [ʃ p r a x] as [ʃ p r ɛ ç], ignoring what comes before the slash (/). While this does not reflect the historical reality of the derivational process, it is a convenient shortcut by which we can handle those cases of non-concatenative morphology which can still be aligned. As you can see from the figure, EDICTOR will render the source/-part of the source/target representation as superscripts.

**The Morpheme Glosses Panel in EDICTOR 2.0**

EDICTOR 2.0 greatly facilitates the annotation of language-internal cognates, as shown above, by providing a new panel called “Morpheme Glosses”, which you can open via EDIT -> MORPHEME GLOSSES. This panel will by default take all data which was currently selected (using concept and doculect filters or the “COLUMN = value” filter), and render all morphemes in sorted form, clustering identical forms and sorting them by their cognate identifier. If cognates and morpheme glosses have not yet been assigned, it will render them with questionmarks. All you need to make sure is that your original wordlist has a column COGIDS and a column MORPHEMES in addition to TOKENS. The following figure shows all morphemes for the words containing the lexical root sprech in my annotated German wordlist.
You can see that EDICTOR lists all morphemes with the words in which they occur and additional columns for morpheme glosses and cognate set identifiers. You can also see that EDICTOR treats the derived form sprach in the same way as the non-derived forms. To edit the data, you just click into a field and start to modify the value. To get a new cognate set ID, just type in a question mark or any string, and it will turn this into the next free identifier.

But this is not all what you can do now. In order to segment a word, just click on the token in the TOKENS column in the Morpheme Glosses panel, which you want to split off of the rest. The rest of the word will disappear and you have to refresh the panel to see the newly segmented morpheme, provided your selection still holds.

You can also bulk-edit your data by clicking on the ID fields on the left and marking those entries which you want to edit in the same way. If you now insert a new value in either the morpheme gloss field or the cognate identifier field, this will be assigned to all selected values, as illustrated in the next figure. To unmark the entries, you either click again at them, or you make a right mouse click on any marked entry.
Specifically the bulk editing feature, although still a bit clumsy in application, helped me a lot to annotate all infinitives, suffixes, and infixes in the German test list.

What turned out to be similarly important was the possibility to filter the data in various ways before investigating individual morphemes. Currently, you have three ways to filter the data when doing morpheme gloss editing. You can use the COLUMN = value feature on top-right of the EDICTOR panel that gives you basic information (remember: COLUMN == value will search for the exact value VALUE, while COLUMN = value searches for the column containing the value). You can also filter by languages and by concepts, using the basic filter of the EDICTOR. And you can filter by cognate sets (only one value allowed so far), by gloss (also only one value allowed) or by form. If you use the cognate filter, only exact matches will be accepted. For morpheme glosses and forms, you can also do a substring search, using ^ as the begin of a sequence and $ as the end. Thus, ^ə n inserted to the filter by form field will filter out all forms that begin in ə n, and ^ə n$ will search for exact matches, accordingly. This is illustrated in the following figure.

As a last feature, we allow to distinguish salient from non-salient morphemes, which can be done by right-clicking onto a morpheme gloss, either in the Morpheme Glosses panel, or in the Wordlist panel. In both cases, the morpheme gloss on which you click
will be marked by a preceding underscore, that changes the layout and helps you to distinguish certain morphemes of interest. Detecting the non-salient and salient morphemes later is straightforward, when parsing the data as a Wordlist in LingPy (List and Forkel 2021) or other software of your choice. In our research, we use this feature to mark morphemes that are important for phylogenetic calculations. But the feature has no concrete semantics so far, similar to our source/target construction, so you can use it to experiment. In my annotation of the German data, for example, I tried to roughly distinguish grammatical from non-grammatical morphemes, since I was interested in deriving word families from the data, and word families are cognate words which share lexical cognate material, as opposed to shared grammatical endings and the like.

**The German Dataset**

My datasets originally stems from the Intercontinental Dictionary Series (Key and Comrie 2016, but was drastically modified by correcting many problematic translations, adding phonetic transcriptions, and also refining the underlying German concept elicitation glosses, which were already published in an earlier blog post (List 2020). My original plan was to share this new wordlist officially in CLDF format, making a nice CLDF package with the help of CLDFBench (Forkel and List 2020), but due to time restrictions, I never managed to bring the wordlist to perfection. Instead, I now decided to share the wordlist directly with the EDICTOR, so that those interested in testing the new EDICTOR features for the annotation of language-internal cognates can just open it directly and experiment with the datasets, either by selecting it from the “select remote file” field, or by specifying the file over the URL: https://digling.org/edictor?file=German.tsv. The wordlist is also available in TSV format from the release of EDICTOR 2.0, which you can find in the folder data/German.tsv.

**Next Steps**

Originally, I planned to provide an extremely nicely annotated German wordlist, following the CLDF formats, with examples for tests in Python and hints on how to use EDICTOR. Instead, I only provide a brief overview now, with an half-baked wordlist that still needs substantial editing in the future. This means that I will have to come back to this topic in the nearer future in upcoming blog posts. But already now, I hope that I could trigger the interest of the one or the other reader to test the new features of EDICTOR on their data.

**References**


