

## Data organizing

Exactly how you organize your data depends on the structure of the language, not to mention your temperament, technical skills, and the tools available to you. Some general guidelines can be followed until better ideas about organising data emerge. While a computer is not strictly necessary for doing field-work, it is such a useful tool that I will presuppose that you are in fact doing most of your work on computer. There are ways to achieve many of the same results without a computer, but they are usually a pain in the neck because they involve shuffling lots of paper. This blurb being about the organisation of data, it is assumed that you already have the data either as recorded sound or transcribed data in a notebook.

One of the first things you need to develop is a lexicon for the language, which contains all of the idiosyncratic information you need to know about each word. This would obviously include the word (transcribed), a gloss in English, and grammatical category information (at least ‘verb’, ‘noun’, maybe ‘adjective’ and other categories as seem relevant). It is also useful to include space for freeform comments and related examples, and depending on the structure of the language, you may want to provide another column for ‘stem’ or ‘root’, if words are frequently polymorphemic.

There are two extremes in data-file construction which I will touch on briefly, and having dismissed those extremes I’ll talk about a medium-tech solution which I advocate. The lowest no-tech solution is to just write words in a Word document, sort of as though you were constructing a dictionary. Here is an actual example of what such a database might look like:

**ech-áála.** *n.* cl. 7-8. finger. This word was also used as a traditional measure, equal to the length of the first two joints of the middle finger, approximately two inches. *byáála biná byóóbuzi* ‘four fingers of thread’.

**ech-áála kiséza.** *n.* cl. 7-8. thumb. Literally ‘male finger’. cf. *ekikúmu*.

**-(y)ala.** *v.* spread, make a bed.

**omw-aalábu.** *n.* cl. 1-2. Arab.

**-(y)áláama.** *v.* balance the sails to get propulsion.

**-(y)alama.** *v.* attend funeral and spend the night.

**-(y)alamila.** *v.* mourn.

**ech-aalamo.** *n.* cl. 7-8. funeral.

**ech-ááli.** *n.* cl. 7-8. bird’s nest, especially referring to where they lay eggs.

**-(y)alika.** *v.* make ripen artificially; make nest like bird.

**-(y)alila.** *v.* lay out mats to make a bed for someone.

**omw-aalilo.** *n.* cl. 3-4. grass carpet. cf. *-(y)ala* ‘spread’.

**omw-aalímú.** *n.* cl. 1-2. teacher.

**en-demabaalímú.** *n.* cl. 9-10. unteachable pupil.

**-(y)aambuka.** *v.* cross, ford.

**ech-aambukilo.** *n.* cl. 7-8. crossing place, stepping stones, bridge, ferry.

**ech-aambuko.** *n.* cl. 7-8. crossing place.

The problem is that it is hard to sift the data and extract examples according to some criterion that you're interested in. Suppose you had a lexicon of 400 words, and wanted all of the nouns. With a file organised like this, that would be hard to do.

On the other end of the scale is the fancy database program. Such programs are often difficult to use (require some programming skill), and are possibly costly. If you have such a program and can easily use it, go for it. There is no program which is free, decent, and easy to use which satisfies the needs of the linguist. The intermediate solution (which I use and will use in this class) that works for most people is a Table in a Word file — the example *lexicon.doc* on the webpage has this structure. The file is basically one large table, where each column is a category of information and each row is an item. It is easy to add or delete columns (and therefore, you can insert new information fields quite quickly); it is trivial to sort by the information in specific columns, and you can search for relevant strings found in a specific column (hence, you could look for all cases of “b” in the data column — more useful than getting all of the cases of “b” in the entire file which would include the English gloss. You will notice that I have a number field to the right, so that I can return the examples to the original order of elicitation after doing a sort, and also I've set up two columns in the middle so that they are in the SIL IPA font which would be appropriate for data fields. I assume that I'll need at least two forms of some words (maybe singular and plural) — I can add new columns if I need to. One thing which I think is important is to keep the shape of the lexicon file uniform: each item has exactly the same number of columns and under a column, the margins for one entry are the same as they are for the entry above and below. The existing lexicon document is like that, and you will probably find it a lot easier to work with the data if you keep it that way.

The way to delete a column (why would you delete information? There might be a reason) is to select the column from top to bottom (it turns dark), and cut with ctrl-x. You can zap all of the data in a column by hitting “del” with a column selected (the column itself stays, but it's emptied out). You can insert a range of columns by selecting an entire column, right-clicking the area and getting the right-click-menu which should give the option “insert column”, which inserts a column before the selected column. Another way to do the same thing is to select a column, copy (ctrl-c) then paste (ctrl-v) which gives you two copies of the column; then, hit “del” to empty one of the copies.

There are two important things to remember when using tables. First, the physical structure of each row should be the same — the same number of columns and positioning. If they are not, you may get confused about what you're doing, and may not be able to manipulate the table easily (especially if some rows have different numbers of columns). Second, each column should serve a single function, e.g. “gloss”, “word class”, “gender”, or whatever.

It is best to put more important columns to the left and less important ones to the right (which need not even be on screen much of the time). The latter category would include “arbitrary coding” information, for example the 9th column might indicate how many syllables in the stem, or the 12th column might mark whether the word ends in a consonant or a vowel.<sup>1</sup> You will often want to print some version of the wordlist, sorted perhaps by word class, but you probably don't need to print out the “order of elicitation” field. By putting the important information first, you can sort the database according to some field off to the right, but you won't actually have to print that field (select the leftmost N columns and print just the selection).

The second most important thing to have, by way of organizing data, is a daily log, where you summarize everything you learned in the day's sessions. Often this takes the form of seemingly unin-

---

<sup>1</sup> You might be able to automate the entry of such information, but it's usually simpler to just type it in for each entry, once you know you really want that information on a regular basis.

formative statements like ‘continued looking at hortative aorist negative subjunctive passive’, but if you mark the date in the log and correspondingly mark the date in your notes, it makes it easier to actually find the examples of the data in question. This is a good place to enter hypotheses, such as ‘it seems that the imperative is marked with the prefix *pəŋ* which loses its nasal before some consonants’; next time, you’ll focus on which consonants do it (and which seem to be counterexamples, which will force you to reconsider the data and the hypothesis in the next session). Especially when you start to amass a lot of data on one particular topic, this may spin off into a topic-specific log file. You should create a log file by the end of the second week, and at the very least you should say what appear to be the sounds of the language, and whether there are any rules about when they appear or what they sound like (because a transcription may not tell you everything that you need to know).

The main point of constructing a daily log file is that it forces you to review the data in an undistracted fashion. When you gather the data you are thinking about how to transcribe the example, trying to remember what the speaker just said so that you can write it down, also thinking at a low level about what the significance of the example might be and what the next question should be. It’s a waste of everybody’s time to hold up the proceedings while you contemplate the significance of *nθpšmka?* as the plural of *λ’ob*, in terms of your current theory of pluralisation. During elicitation sessions, you may think that you understand why the data is one way and not another, but you really don’t have enough time to think deeply about the other kinds of examples that would properly test your current hypothesis. In writing a log file where you include examples of the claim du jour, you may discover that you failed to get any examples of cvvcvcv nouns, or passive agents before time phrases, and discovering this gap should inspire you to gather such examples the next time.

The third most important thing to develop is sets of files focusing on major structural elements. Initially this will be hard to do, because you will have no idea what are the relevant areas. Such a file will contain examples of the structure in question, analytic comments, and especially paradigm and morpheme tables. Generally, one starts with relatively broad files such as ‘verb tenses’ and subdivides when it becomes necessary. Supposing that the language you are working on has many types of pronouns for main clause subjects, subordinate clause subjects, objects, possessives, switch-reference and various other functions, it’s unreasonable to expect to have the whole chart memorized right away. With a handy reference sheet which compactly presents the entire pronominal system, you could glance at the sheet when you’re eliciting structures with “want” to be sure that the form of the pronoun which you’re getting is what you expect.

The unifying principle behind these last two tools, the log-file and the reference sheets, is that efficient and effective fieldwork involves a fair amount of preparation and awareness of the consequences of data turning out one way vs. another. When a surprising datum comes in, it is best that you realise that it is in fact surprising, when you gather it. This may not always be possible, but that is the goal towards which you should be working.

In the most extreme case, every elicited example would get entered into a computer database so that you could trivially extract dozens of cases of X with a few keystrokes. Since it generally takes almost twice as long to enter examples as it takes to elicit them, and since database technology for manipulating field notes is still pretty primitive, (and since presumably you are taking some other class) it’s unlikely that you will actually computerize all of your data. But do try.

A propos computer usage, try to organise data into a hierarchy, so you know where things are. This is the organisation that I use on my computer (folder names underlined). Some of these might be unnecessary for a smaller project, but structuring data is easier when done early.

Data

Somali

Recordings

2301

2801 (etc)

Papers

Texts

Phonology

Syntax

Morphology

Lists

log.doc

plan.doc

lexicon.doc

(all of my field notes)

(this language)

(digitized materials)

(papers I'm writing on the language)

(stories and the like)

(things like minimal and near pairs, summaries like what rules I think there are, more extensive paradigms that I might use to fully justify e.g. a palatalization rule )

(similar)

(similar; especially include paradigms)

(This is usually a specialised variety of "lexicon.doc"; for example, one list might be "Verbs sorted by English gloss", another might be "all words sorted by Somali form", or "Somali nouns", or "monosyllables")

(the running commentary file)

(a repository of questions to ask next)

(the main lexical database)