Practical Prerequisites (LIFT Database File to Check)

In case it isn't obvious, in order to use this program to actually do anything, one needs a LIFT database to check. Fortunately these are not hard to generate; LIFT is an open XML specification for storing lexical data. You can create a LIFT database by a number of routes:

- Collect words in WeSay (Download here).
    - WeSay uses LIFT natively, so the same repository can be used for WeSay and A→Z+T (though I wouldn't recommend you have them both open at the same time).
    - WeSay is Chorus enabled, which allows easy tracking of changes and off-site archiving, including the changes to your database made by A→Z+T.
    - While you're installing things, I highly recommend the excellent library of images that works with WeSay, the Art of Reading
- Build a dictionary in FLEx, and exporting to LIFT/WeSay.
    - assuming you want to get the checked database *back* into FLEx, you want to use send/receive for WeSay. Import/Export (a different process) can work, but there is no protection against overwriting your data, without doing back-ups yourself.
    - there is an active list of users to help with problems doing this.
- Store your data in some other form (text, spreadsheet, database) and convert it to LIFT (*PLEASE* don't do this unless you *really* know what you're doing, and have a *good* reason to; the other options above are much easier, and much less likely to result in data corruption)

Collaboration and Archival

I **strongly** recommend using a version controlled repository (e.g., mercurial, git), as is normally done in WeSay and in recommended FLEx collaboration schemes. Even if you are the only one to ever see this data (why would that be?), the advantages in history and preservation of your data are already there. But if you will be sharing changes with others, you really **must** have an easy way to do this, or you will get bogged down in the logistics of sharing data changes. I recommend using Language Depot, though there are certainly solutions for this need on other servers. In any case, setting this up early is always easier than trying to merge divergent data later.

LIFT Database Requirements

LIFT databases can be minimal or very complex. For the purposes of running A→Z+T, you just need the following:

- `citation` or `lexeme` forms (tagged with your language code, of course)
    - forms with spaces or other non-wordforming characters are ignored.

- `glosses` or `definitions` in at least one language (again coded for gloss language)
- Grammatical Category/Part of Speech (ps) indication:
  - stored in `sense/grammatical-info@[value]`
  - whatever ps names (e.g., 'Noun', 'Nom', 'Njina', 'noun', 'n', etc) are in your database is what you will select from to study, so name them appropriately for your work
  - entries with no ps value will be left out of the A→Z + T analysis

Changes to Expect in Your LIFT Database

A→Z + T will place links to `citation/lexeme`, `plural`, and `imperative` recordings in the appropriate fields, coded as a voice writing system (e.g., xyz-Zxxx-x-audio).

A→Z + T will place sorting information in the `entry/sense` node, under an example node for each context/frame:

- form
- sound file name
- translation
- frame name
- subgrouping name

Results of the analysis of multiple frame groupings (e.g., from the Tone Report) is placed in a separate `entry/sense/field[@type='tone']`, as this is a summary/analysis of the values contained in the example nodes.

Changes to Expect near Your LIFT Database

A→Z + T assumes your lift file is in a directory set apart for its analysis. As a result, expect to find generated in that directory:

- A syllable profile analysis file (on first run, and if CV analysis parameters change). This can take a while to run, so we store and load it, rather than running it on each startup.
- A preferences file (any time a check is run, to preserve preferences, including `ps/profile` status). This allows you to start where you left off.
- A tone frame definitions file, once at least one has been defined. I strongly recommend not modifying this file.
- An `audio` folder, where sound files are saved, once at least one has been recorded.
- A `report` folder, where report files are stored whenever run by the user. These are set apart from the rest of the repository to reduce clutter. N.B.: relative links to audio work in this hierarchy, if you move a report from a sister directory to `audio`, be sure to update the links accordingly.

Tell A→Z + T Where to Find Your Database

The first time you run A→Z + T, you will need to select your LIFT database.
A→Z + T stores this location in `lift_url.py`, so you only have to do this once. But
if you need to check a different database, delete `lift_url.py`, and A→Z + T will ask
again where your database is.

First Run: Be Patient and Orient Yourself

Syllable Profile Analysis

If you open A→Z + T without a saved syllable profile analysis file, it will first open your
database (once it knows where it is) and go through the entries there, and sort them
by syllable profile and part of speech (CVC v CVCV for each of Nouns and Verbs, for
example). This can take a couple minutes. If you have a terminal open, you should see
its progress.

Configuration File

If you open A→Z + T without a saved configuration file, the program makes assumptions
for you (based on your LIFT file), so you can get started right away, if you want to, e.g.:

- the largest syllable profile group, and its ps
- vowels first, before doing consonants or tone
- the segment with the largest representation in the database is chosen first.

If you don't like those assumptions, you can change any of these settings in the menus.
Those settings are saved to a configuration file each time you run a check, so your
preferences will be there for your next run. If you change a setting that leaves another
setting invalid (e.g., `V1=g`, or `C1=C2` for `VCV nouns`), the invalid setting is removed and
replaced with an assumption as above, until you change it.

The Main Window

On the upper left of the main window, each of these settings are indicated:

- interface language
- analysis language
- gloss language(s)
- part of speech
- syllable profile
- type of check (i.e., C, V, or T) and current stage

The main window menu allows each of these to be changed as needed.

Tone Frames

Assuming you don't have any tone frames set up yet, you will be asked to do so when you try to sort on tone. You can also do that in the Advanced menu, for as many frames as you want to define. Note the *name* is important, as this is how you will refer to this frame, and how it will be identified in your database in the future (until/unless you change that). So if you're testing the plural form, something like "Pl" or "Plural", or "Pluriel" might be appropriate –but this is just a name, so make it distinct but useful to your workflow.

I hope the Add Tone Frame window is otherwise clear, though two points are in order:

1. I have seen on at least one MS Windows system, that keyboarding that takes multiple keystrokes to produce a character (like 'n'+'>' → 'ŋ') may show up as '?' in the entry field. If this happens, *do not* ignore it, as this will be on every window, and added to your database examples. Rather, type the correct characters into another program (e.g., text editor), then cut and paste into this field, and it should appear correctly.
2. The frame calculator is not particularly smart; **it just puts content before and after the form and one or two glosses** for each word, so you need to give it that information. If that information (in the form or gloss) alternates in agreement or harmony with the lexicon word forms, you should think through how you want to resolve potential clashes, e.g., by

   • including all options in the frame:
     – `pl form: '__s/z/ɪz'` (with all forms given for each word in that frame)
   • or allowing for ungrammaticality:
     – `pl form: '__s'` (knowing that dʒədʒ+pl will come out 'dʒədʒs', not 'dʒədʒɪz')

The user will have the option to skip a word that doesn't work in a frame. This was put there for syntactic or semantic clashes, but could be used to exclude phonologically weird frame combinations, too.

Once you have the form and gloss content in the appropriate boxes, click on 'see it on a word from the dictionary', and you will get the frame as you have defined it applied to some word (in the filter you have currently set). You can try this on a number of words; **continue to click that button**, to see how it will look on different entries. There is no easy way to change this frame after you define it (other than deleting them all and starting from scratch), so I encourage you to take your time before moving on. When you are happy with the frame, click on "use this tone frame".

If you absolutely regret a tone frame you have set up, all your frames are stored in `<lift filename>_ToneFrames.py` next to your lift file. Careful editing this, though; you may need to redefine all your frames if you corrupt this file (This would be a great time to ask for help if you don't absolutely certainly know what you're doing).

Recording

The first time you try to record, you will be asked to tell A→Z + T what sound card parameters you want. You can set frequency, bit depth, and sound card number (to select between multiple cards). This window is designed with a test button, so you can set parameters and test them, before moving on. I suggest you budget some time to play with the settings there, and confirm that recording is working, before moving on. I suggest you select the highest quality that your card can do (assuming that's what you want!), and test to see if it records and plays back OK. I have found several computers with cards that can record at 96khz, somewhat to my surprise —though be sure to think about your microphone and environment, etc, too! If you are making recordings for easy sharing over low bandwidth (as opposed to linguistic study), consider the implications of these setting on the size of your files.

The sound card settings should be stored in your repository, so you won't have to keep setting them. However, if you are sharing a repository across computers with different recording capacities, you will want to pay attention to this; the sound card settings dialog is also available through a menu item.

Subsequent Runs: CV analysis (View data and run reports)

A→Z + T doesn't do CV sorting and verification (Yet!), but you can make recordings and filter your data and look at it through a number of checks (e.g., by C1, or by V1 = V2, etc.).

Recording Citation and Secondary Forms

When Consonants or Vowels are selected, you can click on "Record Dictionary Words", which will give you a page of `Record` buttons next to words filtered by ps-profile combinations, largest first. To skip to the next slice of data, just click "Next Group". For each entry in the data slice, this page provides a button to record a sound file for citation or lexeme fields, but also plural and imperative fields, if in the database. These recordings should then appear in reports, FLEx, and other uses of the LIFT database (e.g., the Dictionary App Builder).

Consonant and Vowel Reports

To look at filtered data, set up the desired parameters so they appear on the main screen, then click "Report!". This results in a window with data to scroll through, and an XLing-Paper XML document with the same output, ready for printing to PDF or HTML through the XMLmind XML Editor (XXE).

For a more thorough report, use the Basic Report menu item. This will select your top ps-profile combinations, and present each of those slices of your data in order, with the relevant checks for consonants and vowels for each. So the CVCV profile will start with C1 = C2, before moving to other data sorted by C1, then by C2 —then it will do the same for vowels. Longer syllable profiles will thus include more checks, though maybe not containing much data (e.g., if you don't have much C1 = C2 = C3 data).

I want this tool to be ultimately able to help with the sorting and correction of consonants and vowels, but my hope is that until those functions are implemented, these reports will be helpful.

Subsequent Runs: Tone (Sort, and Follow Directions)

Sorting progression

Once you have done any sorting for the selected part of speech, to the right on the main window you will see a status pane, with groupings by syllable profile and check stage (for one part of speech and check type at a time). To see progress for another check type or part of speech, switch to that check type or part of speech.

The program is designed to step through the process relatively automatically; once things are set up, you should be able to just open the program, and click Sort. If you need a break, click quit on whatever window you're in, and your progress should be there when you return.

You will, of course from time to time want to move to another part of speech or syllable profile, or check type. That can again be done on the main window menus, and the next time you click Sort, the appropriate checks will be launched, and those changes saved to the preferences file.

Recording Data Sorted in Frames

Recording data in frames can be done at any point where at least one frame has been at least partially sorted, but when a word is presented for recording, each example (sorting context) is presented for recording. So if you sort one field, then record, then sort the next, you will see your earlier recordings again. Recordings seem to move rather quickly, so I recommend putting them last in your workflow, and do them all at once —at least once you've tested that they're working correctly with your sound card, etc.

Tone Reports

Once you have done some sortings, it makes sense to run a report. The tone report will show your groupings in just one frame, if that's all you have done, but its real value

lies in comparing values across multiple frames, so you'll want to check a couple tone frames before doing much with the tone report.

This report is also exported to text and XLingPaper XML files, which has similar organization, but more detail, that what you will see on the report window.

Ad Hoc Sorting Groups

After you have done the analysis of noun and verb roots, you may want to move to a smaller grammatical category, and you may find that the slicing of data into strict ps-profile groups makes it difficult to see what's going on in your data. If you want to manually select a set of morphemes to sort, either a subset of a CV profile group, or including data from multiple CV profiles (or both), click on "Add/Modify Ad Hoc Sorting Groups" (In the Advanced menu, of course!). This provides a window where you can give a (unique!) name to your *ad hoc* group, and select the senses from your dictionary that you want to compare in it.

This window will leave you with that group selected, so you are immediately ready to sort the senses included in it.

Your *ad hoc* group is treated by A→Z+T as another CV profile group, so you can select it from the "Change/syllable profile" menu item, or else move back to a standard CV profile group.

Once you have an *ad hoc* group defined, you can modify it by selecting this menu item with that group selected on the main screen. To create another *ad hoc* group, select a standard CV profile on the main screen before selecting this menu item.

Miscellaneous

If any of the directions are unclear or inappropriate in any way, please let me know so I can fix them.

The UI is currently in French and English; if you want to translate it into another language, please let's talk!