

Roadmap

release stoppers

- put image creation after splash creation?
- give warning if user is going to undo analysis regroupings
- check out why tooltips aren't working before first refresh

Documentation

- Steps to use AZT in a workshop, from start to finish (not absolutes, but recommendations)

Things to Test

- check that changes to S regexs don't break too much.
- Why do bad sound files play OK?
- figure out why mainloop is still running on window closure
- method of speeding up image scaling:
 - CPU detection, set y in xy/y function according to processing speed.
 - multithreading
- make font changes more general, tied to +/-

Simplify (non-OOP related)

- on import, check for entries without:
- citation, copy over from lexical-unit (all langs)
- gloss, copy over from definitions (truncate, all langs)
- remove references to definition, lexeme?
- Do I want to ultimately be modifying lexeme form?
- check how many of the urls in lift.py I'm actually using, consider moving them to a method for each —
- maybe call get with a url parameter?
- kwarg for node or not, text or not, attr?
- find ways to speed up tone reports:
 - multithreading? !(CPU limited)
- make dictionary of images created with PIL, so they aren't continually remade (source if already there)
- distinguish between users and their functions:
 - language:
 - * sort, etc.
 - * record

- * make reports
- linguistics
 - * transcribe (but not sort, etc)

Migrate further toward OOP

- mark a sensid to sort again without losing the sound file attached to it.
- distinguish between frames to do for sorting (with unsorted data) and frames to do for other tasks...
- make 'next' go to next frame, if done sorting, or not, as appropriate
- make tone analysis one thing, and tone report another thing, and call analysis if not done since data was last added/modified (store this info somewhere, clear after analysis completes)
- set report for different forms separately (e.g., making images for screen may not be useful for some, yet takes time)
- Consider making the following objects?
 - fn to process kwargs w/default (currently hardcoded in buttons/labels,etc.)
 - ErrorWindow class (tryNAgain, maybesort ...)
 - tables (base on buildXLTable(self,parent,caption,yterms,xterms...))
 - languages
 - ps-profile slices
 - check name - frame
 - sense (create these objects in threads)
 - group
 - runwindow
 - groupselected should be an attribute of a sense?
 - lift-form? include attributes lexical, citation, root, affix(es)?
 - this will need some treatment by language, for dictionaries with forms in multiple languages.
 - lift-field?
 - lift-node?
 - lift-example?
 - status dictionary:
 - * methods to check done, there, etc, without having to check whole dict.
- Think through guessing and next functions, and maybe verify?
 - Sometimes guessing is appropriate, especially current is not valid
 - * and not necessarily the same as next (i.e., not current)
 - * nor verify, which makes valid, but doesn't change if already valid.
 - These should probably all go together, with common [arg,argprioritylist] parameters
- Consider when/where it makes sense to do multithreading, if it is possible to generalize it in a class

In Process (fix)

- go to <https://fontforge.org/docs/scripting/python/fontforge.html> to see if we can pull contour tone staves
- add mini keyboard of common symbols (class, to add to Entry)
- fix to change tone frame name, everywhere where needed
 - toneframes.py
 - verification file
 - example nodes
 - verification nodes
- make verify window go away when last button is clicked
- Set up schema to test line expenses on boot time, for each line
 - this-then, store then = this
 - determine costliest lines, which can be threaded
- Set up schema on largest (time) expenses, to do any of (to test real world difference, as well as to have available for quick changes, if needed later):
 - as is
 - threaded
 - multiprocessed
- freeze header on status scroll (not currently practicable, without serious reworking of the table/scroll)
- don't write ps = 'Invalid' to profilesbysense
- make settonevariablesbyprofile take arguments ps, profile
- Frame object has no attribute 'skip'
- reduce unnecessary self.gettonegroups() calls
- make "next frame" do checkcheck
- test for table content before starting it; if all zeros, skip
- Don't die on no count for table
- fix analang detection problem?
- make recordbuttons pass recordcheck, which verifies that settings are plausible.
- include checks on empty or repeated XLP nodes
- convert basic report to XLP table function
- set up function to change tone sort group names (to transcriptions/meaningful names)
 - give buttons for each tone letter, plus back and enter
 - give scrolling frame of buttons for framed words in the group, clickable to play to make the transcription
- make status table scroll (in both directions?)
- Look at tkinter tabs (for status page?)
- add progress of recording (on its own tab?)

For Future Versions

- Add comparison group button for rename framed group window (to help with transcriptions)

- consider adding actual groups by datapoint for non-default tone reports
 - I.e., which data points are different
- think through ease of use questions:
 - How to make naming groups straightforward from the main interface?
- implement XML2XLP.txt (to produce PDF without further user input)
- add interpretation of glottal stop into sdinctions.
- fix zip problem on Windows: “OSError: [Errno 22] Invalid argument: ‘log_-.7z’”
- extra space being added for None forms in Frame construction
- Sort out why not all nouns show in ad hoc selection list.
- functions that depend on a slice of data:
 - require renewal of certain variables before running
 - careful iterating over them
 - distinct from other functions, which iterate over multiple slices
- make nextps and nextprofile (and nextframe?) reference most popular *unfinished* ps-profile combo
- fix skip button
- reduce calls to gettonegroups; just after check name is set
- confirm that all variable calls that are only set on frame/ps-profile switch are stored, and available on open.
- don’t write blanks to verification file
- for next ps/profile, move to modified settings (if there), rather than re-figuring each time (this costs nontrivial time).
 - if nothing has changed, there’s no reason to waste that time.
 - we can refigure once per ps change? To hold it longer we would need to store it in a dict keyed by ps.
- propagate exitFlag to the rest of the check
- better document skip function
- improve status table with buttons
 - make ‘next’ buttons on the end of each axis in the table.
 - make “hide” buttons for each header (x and y)
 - make “show all” button at the origin
- commit diffs to hg on closing (or when?)!
 - think through which settings should be shared:
 - * profile data?
 - * verification status
 - * ad hoc groups?
 - and which shouldn’t
 - * check defaults
- fix windows minimization problem (new with availxy)
- consider adding ps = None, profile = None, and name = None to scripts used in iterating across these variables, so they can be sent as variables, rather than changing self.{ps,profile,name}, given that these changes will now come at a greater cost.
 - all changes to self.{ps,profile,name} should come through self.set()
 - avoid unforeseen effects of self.x = y calls
 - We need to distinguish between user changes to these variables, and changes from iteration under the hood, which should be through arguments of local

variables.

- search for any script wrapped by `psori = self.ps` type statements, amend.
 - * `settonevariablesbyprofile`
 - * `updatestatus`
 - * `makestatusdict`
 - * `makeadhocgroupsdict` (done)
 - * `sortingstatus`
 - * `gettonegroups`
 - * `getframestodo`
- move all ad hoc changes by attribute (e.g., `self.ps = x`) into `set` (to generalize it)
- Make function to manually refresh status file, don't do it otherwise
 - this is getting cumbersome, like the profile data
- Remove None from status list of profiles –this should never go there.
 - define `self.profile` before writing to status?
- Reduce `checkcheck` calls, and/or separate table refresh from northwest status?
 - or make `sorting` add to stats files, so we don't need to recompile them but when changing `ps-profile` or frame
 - make wait window work in the mean time
- Find out how Chorus decides what files to pick up, make sure our config files are getting in.
- move config files to `aztconfig` directory?
- look at how to generalize tone sorting, joining, etc. process.
- bring new documentation pages:
 - What A→Z + T will and won't do for you
 - The Cyclical process of analysis
 - * incremental
 - * complete cycles
- get dependencies offline for `pyaudio`
- don't crash on invalid sample rate; clear if not valid.
 - use `is_format_supported` on parameters, reset, or limit options?
- When making record button:
 - remove links to the *wrong* sound file
 - test for presence of currently linked file (i.e., not recorded in AZT), give play buttons
 - widen “Do” menu item (give all a minimum width)

For some time

- set up mail of bug report (better than WeSay)
- look into having hg commit changes to verification status file
 - don't track `checkdefaults`? (make per user file?)
- Set means for user to check verification stage again.
 - Once done, there is currently no AZT way to redo it.
- Look up how to get real required heights and widths, `availablexy` isn't working correctly.

- fix reconfigure scrolling window frame problem (remove need for if self.configured < 1:)
- constrain frames with less data, to only scroll as needed.
- add test for resolutionsucks, implement smaller font theme
- include test to see if a tone analysis has been run since latest triage;
 - if not run the report
 - For verification window, if last example is selected (all gone), exit
 - * don't ask for nonsensical “these are all the same”.
- ways to change a frame after the fact... this seems to be a common problem
- ?put next undone buttons in dictionary recording pages, too
- for addframe, add button/menu to allow viewing all frames already done, to help with consistency across frames.
- ?cycle through tone groups to record based on volume of the group.
- done? Update regex functions a= to allow for C(V)C\1, CVC(V)C\1, and C(V)C\1C\1, for vowel and consonant reports and checks

Next Features

- add C and V sorting (and CV?)
- model form construct with fields for CV info and tone info, which may contain variables (HAB,CONT) in addition to H, L, etc.
- proper modeling of the relationship between lx and lc
 - show pronounceable words only
 - build words on the basis of roots —should some frames use lc, e.g., if affixes are unpredictable?

Some Day, if possible

- transition to gloss only (no definition references)
 - make docs specify gloss should be populated (maybe instructions to bulk copy?)
- Look into tkinter tabs to separate out different task sets, like
 - sorting v recording
 - PS1 v PS2
 - C v V v T
 - this ps-profile slice, v whole database work
- rectify (scrolling) Frame redundant window size fn
- have automatic choice of ps-profile go through all defined frames, then next most populous ps-profile
- include tone grouping in recording pages?
- XLP export (these are not likely to happen)
 - run dotexpdf (once you figure out where that is) to generate pdf?
 - run script to generate html?
 - figure out window specific variations (with Andy?)

- set tone report to ignore NA, if there is only one other value in that position (given the other values for the other frames).
- Fix 'η' display problem in entry fields in MS windows (working fine in Linux)
[x] noted issue in USAGE.md
- Make Wait window show background color and label message in MS Windows (working fine in Linux)
- put setdefaults.py into Check class
- distinguish between lc and lx
 - make CV report only reference lx field
 - make docs specify the difference, start with lc references (maybe instructions to bulk copy?)

For specific databases

- fix key 7 problem for sxw
- ? figure out why multiple funj entries aren't showing for recording on bfj
 - '6e2a67cb-6695-4536-bc55-423fad4f019b', <+
 - '7cdcddfc-5b9f-44bf-bfe5-d5ce164720cd',

Documentation

- Add what and why pages in different places, with rationales and instructions specific to context?