Instruction Violation Detection Script

About

Experiment that presents an audio stimulus which is a sentence. The sentences are correct or incorrect and the participants is to hit a button when the sentence is incorrect. An optional yes-or-no comprehension question might be asked following a sentence. A progress bar is shown during a trial that signifies how much time is left for the participant to respond.

Starting Violdec

1. Make sure that Zep 1.10 is installed.
2. Unarchive the .zip file to a folder known to you. (It might be that the file-extension reads .qqq, in that case rename it to .zip and retry.)
3. Navigate to the folder you just unarchived the script files into.
4. Either open the linux-terminal.sh or windows-terminal.bat depending on your operating system of choice. If you are very worried about reaction-time accuracy we recommend a real-time Linux kernel and a button box (ask tech support).
5. Enter 'zep violdec.zp' at the terminal that pops up and a new menu should appear.
6. Click on *Participant* and when you have a new participant click on *New* otherwise navigate to the correct participant id and jump to step 8.
7. Enter a participant id (name, or perhaps something a bit less private).
8. Press *ENTER* and then hit the button *Start*
9. The group can be ignored; press *Continue.*
10. The script will now start.

Configuring Violdec

Most of the timings and pauses can be set in *modules/global_defss.zm.* Be sure to read the corresponding comments before editing any of the options!

Texts that are shown can be edited in *modules/texts_en.zm.* Note that you can use basic HTML for highlighting (bold, italics, <u>underlined</u>.

Fonts and colors can be edited in their respective default files *modules/fonts.zm* and *modules/colors.zm*. Note that there are a bunch of additional settings which are not relevant for the script. Check the corresponding 'pages' on which colors control the looks.

Simuli are defined in the .cvs files in the *stimuli* folder. Sound-file names are expected to be formated as the ID of the stimulus with the file-type extension as suffix (*i.e.* '*1.wav*' for stimulus with ID 1).

Making sense of the output

All the output is set into the 'db' folder. You can edit/retrieve the data by navigating to the right folder within the hierarchy or you can run 'zepdbextract' (via the terminal, more info can be found via the ZEP reference manual). The latter command collects all the session-by-session data and generates two summation files. The first file ('violdec-*01-1.cvs*') contains the results of the test phase whereas the second file ('violdec-01-2.cvs') contains the results of the practice phase.

OUTPUT	DESCRIPTION
dbid	Database-identification
expid	Experiment-identification
resid	Researcher-identification
ppid	Participant-identification
grpid	Group identification
sesid	Session identification
sesdate	Session date
sestime	Session time
sesrepexp	session repeat counter for experiment (how many times has this participant done this experiment before)
sesrepgrp	session repeat counter for group in experiment (how many times has this participant done this experiment before while assigned to this group)
sesstat	session status: 0=running, 1=stopped, 2=aborted, 3=finished, 4=failed
expstat	experiment status: 0=preparing, 1=piloting, 2=testing
expver	experiment version (experiment defined)
trialnum	The number of the trail
id	Identification number of the sound file and sentence
test_sentence	Actually sentence that was read out
type	If the item is a CRITICAL or FILLER type
gram	Grammaticality correctness of the sentence
comp_question	A(n optional) comprehension question
question_answer	The correct answer to the comprehension question
rt	Reaction time measured in ms from the start of the sentence playback (-1 = missing)
correct	If the response was adequeate (0 = no, 1 = yes, -1 = missing).
rt_question	Reaction time measured in ms for the comprehension question (-1 = missing)
correct_question	If the response to the comp question was adequeate (0 = no, 1 = yes, -1 = missing).

DISCLAIMER