Concealed Information Test application

May 03, 2019

Author: Gáspár Lukács

Contact: lkcsgaspar@gmail.com

Below you find the documentation of the of the Concealed Information Test application (v1.1.0), by sections *Requirements and usage instructions, Introduction, The CIT versions, Task design, Settings, Results, Evaluation, Planned features, Support, Citation, References*.

Requirements

This application needs no installation whatsoever, it can and should be run in a *Google Chrome* browser (v25 or above). There are no further dependencies or system requirements. The entire application works offline: no internet connection is needed.

The easiest way to use the application is via the site of the repository itself. (Note that no data is saved or sent anywhere on the server. In fact, once loaded into the browser, the test may again be used entirely offline as well.)

However, the application can of course also be downloaded or forked and/or cloned via Git. This has three advantages. Firstly, the application will not be affected by updates. This may be desired when it is used for an experimental study where the interface and all settings are preferred to remain identical throughout the data collection. Secondly, the application will always be available offline (in case of downloading it). Thirdly, the code can then be modified for specific purposes. For example, in a simple case, the experiment title and settings may be given new default values to be used in a series of tests of a study.

If the application is downloaded on a PC into a local folder (including all directories and files in the repository), it can simply be opened in Chrome (e.g., by dragging the main *CITapp.html* into a new window or tab of the open browser). Otherwise, these files can also be uploaded to a server to make the test available online. If the application is forked or copied into a new GitHub repository, a webpage for the application may also be easily created using GitHub Pages (available under repository Settings).

Introduction

No deception detection method has been scientifically proven to achieve perfect or close to perfect accuracies in discriminating deceptive ("guilty") persons from truth-telling ("innocent") ones, notwithstanding the claims by some commercial services and products. The most efficient "Concealed Information Test" version in the present application ("enhanced") achieved remarkably high accuracies (*Lukács, Kleinberg, & Verschuere, 2017; Lukács & Ansorge, 2019*), but even so it can only be said to provide an accuracy of roughly around 90% – under certain conditions – and it still needs further research and replications. Correspondingly, this application is only for testing and research purposes, and no serious judgments should be based on its results. Furthermore, anyone using it should have at least a basic understanding of how it works – see below.

The CIT versions

This application implements three versions of the reaction time-based Concealed Information Test (RT-CIT; *Seymour et al., 2000*; *Lukács, Kleinberg, & Verschuere, 2017; Lukács & Ansorge, 2019*): *standard, enhanced,* and *no-target*.

The *standard* and *enhanced* versions aim to reveal whether or not a certain information detail – the *probe* – is known to the tested person. **In these cases, the identity of the probe should not be shown to the person taking the test: the starting page must be first completed and submitted by the person who administers the test. The *irrelevant* and *target* items must be similar to the probe, and indistinguishable for a person who does not know the relevance of the probe (e.g. the probe is a stolen suitcase, and the irrelevant and targets are other suitcases).**

The *no-target* version is a tentative approach to make the test applicable in cases where the *probe* is actually known to the participant. This version still needs a lot of research to see how and to what extent it may be applied. For details, see the paper by *Lukács & Ansorge (2019)*.

<u>Standard RT-CIT</u>: it has one probe, one target, and four irrelevant items. The target item will require a different response in the task, otherwise it is the same as an irrelevant item. A person who knows the relevance of the probe (e.g., because he stole the suitcase) will tend to make slower responses to it than to the irrelevant items.

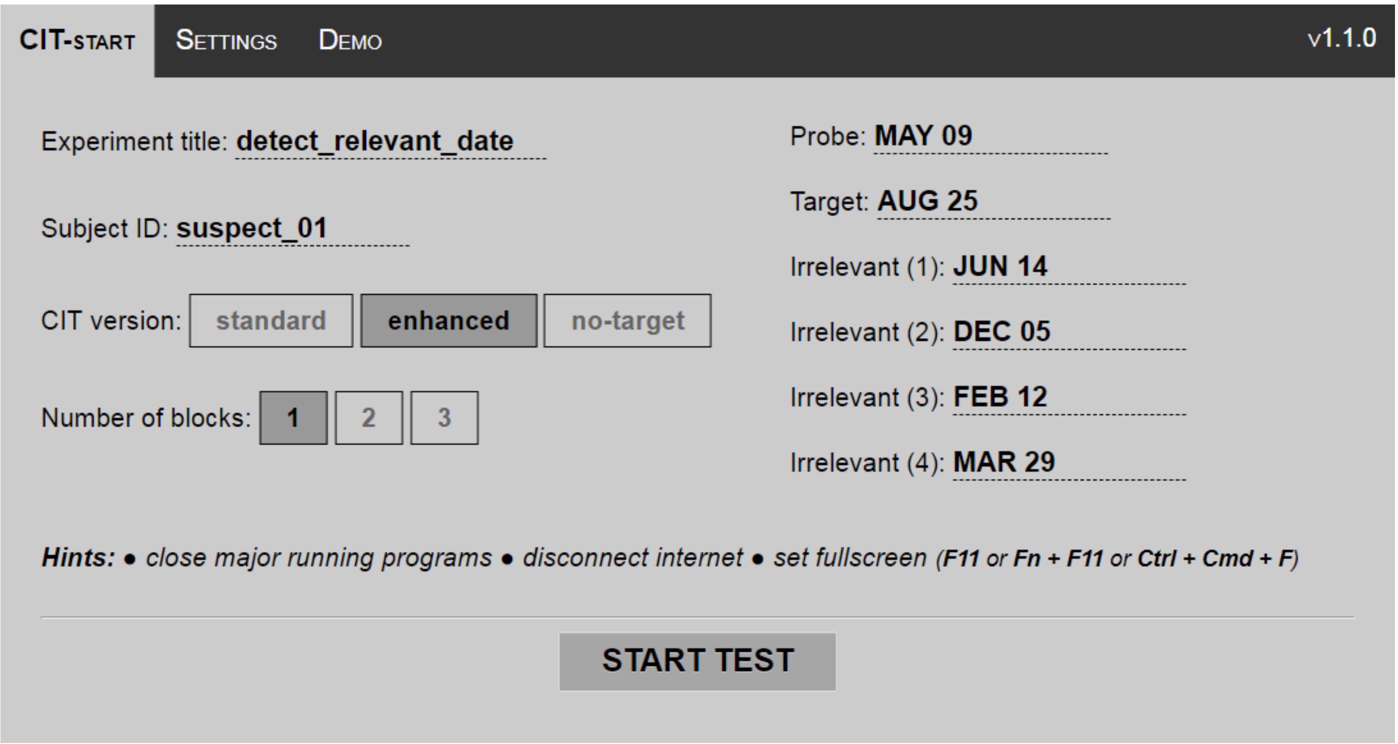
In the standard RT-CIT, participants classify the presented stimuli as the target or as one of several non-targets by pressing one of two keys. Five non-targets are presented, among which one is the probe, which is an item that only a guilty person would recognize, and the rest are irrelevants, which are similar to the probe and thus indistinguishable from it for an innocent person. For example, in a murder case where the true murder weapon was a knife, the probe could be the word “knife,” while irrelevants could be “gun,” “rope,” etc. Assuming that the innocent examinees are not informed about how the murder was committed, they would not know which of the items is the probe. The *items are repeatedly shown in a random sequence, and all of them have to be responded to with the same response keys, except one arbitrary target – a randomly selected, originally also irrelevant item that has to be responded to with the other response key. Since guilty examinees recognize the probe as the relevant item in respect of the deception detection scenario, it will become unique among the irrelevants and in this respect more similar to the rarely occurring target. Due to this conflict between instructed response classification of probes as irrelevants on the one hand, and probes' uniqueness and, thus, greater similarity to the alternative response classification as potential targets on the other hand, the response to the probe will be generally slower in comparison to the irrelevants, and thus, based on the probe-to-irrelevant RT differences, guilty examinees can be distinguished from innocent examinees.*

Enhanced RT-CIT: in addition to the items in the standard RT-CIT, this has familiarity-related "inducer" items that increase the awareness of the lie detection context, and thereby largely increase the probe-irrelevant difference (even slower responses to the probe).

These inducer items induce a particular semantic context that increases the sensitivity of the method for the deception-related meaning of the probe. In the original study (Lukács, Kleinberg, & Verschuere, 2017), the probes were general autobiographical details (birthday, favorite color, etc.), and, correspondingly, the inducers were familiarity-related: Inducers referring to the category familiar (e.g., "FAMILIAR," "MINE") had to be categorized with the same key as the target and, thus, with the opposite key than the probe (and the irrelevants), while inducers referring to the category unfamiliar (e.g., "OTHER," "THEIRS") had to be categorized with the same key as the probe (and irrelevants). It was assumed that in this way the inducers draw attention to the semantic meaning of the alternative response choices in view of the deception detection scenario (namely, that guilty examinees are denying their familiarity with the probe), thereby increasing the response conflict when the probe appears, making the response even slower. Moreover, the increased cognitive load (due to the increased complexity) also requires more attention throughout the task, which likely facilitates deeper processing of the stimuli. This method robustly increased the CIT accuracy of distinguishing guilty examinees from innocent ones.

No-target RT-CIT: the same as the enhanced RT-CIT, but with no target items. (For details, see *Lukács & Ansorge, 2019.*)

The CIT version version can be chosen on the start page, where the probe, target, and irrelevant items are also to be given.



Experiment title: detect_relevant_date

Subject ID: suspect_01

CIT version: standard [✓ enhanced] no-target

Number of blocks: [✓ 1] 2 3

Probe: MAY 09

Target: AUG 25

Irrelevant (1): JUN 14

Irrelevant (2): DEC 05

Irrelevant (3): FEB 12

Irrelevant (4): MAR 29

Hints: *● close major running programs ● disconnect internet ● set fullscreen (**F11** or **Fn + F11** or **Ctrl + Cmd + F**)*

Task design

The number of blocks can be chosen (1, 2, or 3). One block has the same length and arrangement as in all recent studies (*Lukács & Ansorge, 2019*; except for a slightly extended response window; 1 s instead of 0.8 s). There is no data to show how many blocks per item categories (one probe with corresponding target and irrelevants) are optimal, but, as a rule of thumb, one block should give a fair estimation on the group level, while more than three would probably not serve much additional information even on the individual level.

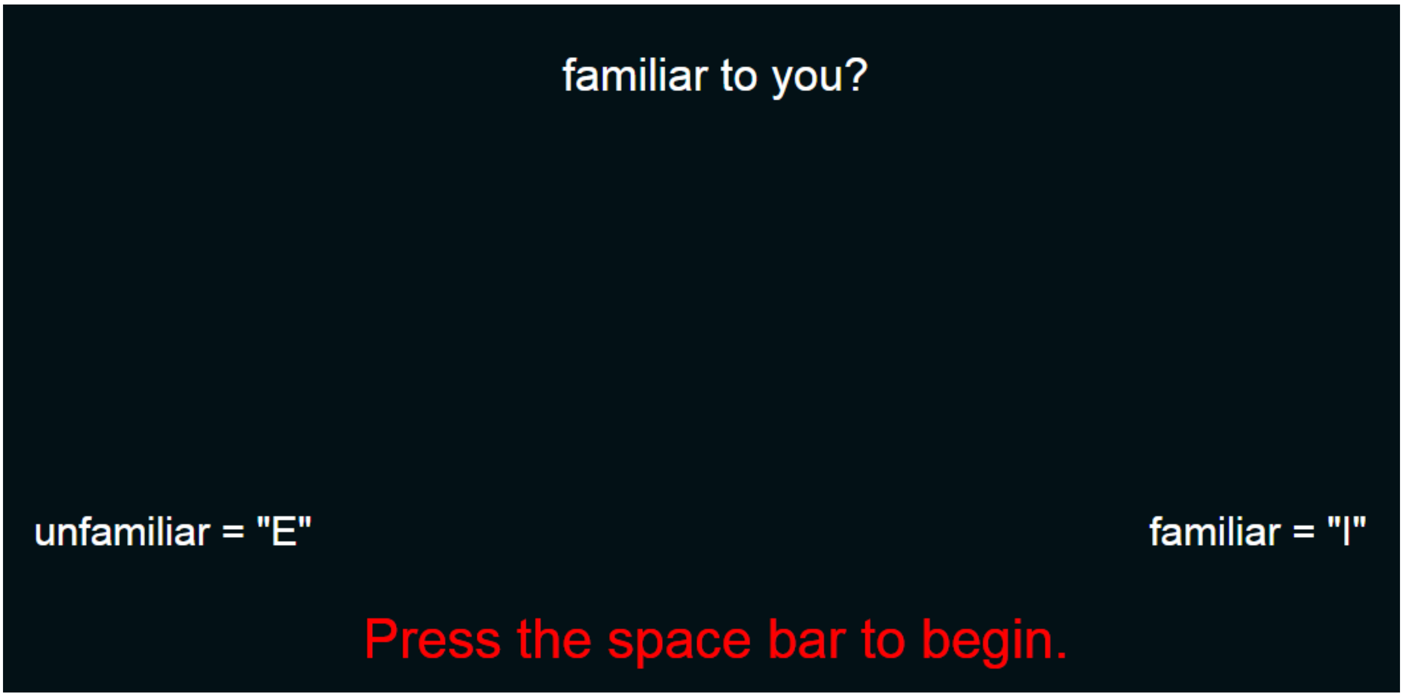
In each block of the standard and enhanced versions, each probe, irrelevant, and target is repeated 18 times (hence 18 probe, 72 irrelevant, and 18 target trials). In the no-target version, there is no target included (hence 18 probe and 72 irrelevant trials). The order of the items is randomized in groups: first, all five or six items (one probe, four irrelevant, and, where applicable, one target) in the given category are presented in a random order, then the same items are presented in another random order (but with the restriction that the first item in the next group is never the same as the last item in the previous group). In the enhanced and no-target version, inducers are placed among these items in a random order, but with the restrictions that an inducer trial is never followed by another inducer trial, and each of the nine inducers (three familiarity-referring, six unfamiliarity-referring) precedes the probe, the target (for enhanced, but not for no-target), and each of the four irrelevants exactly one time, in each block. (Thus 9 × 6 = 54 inducers are presented in the enhanced

CIT, and 54 out of the 108 other items are preceded by an inducer. Similarly, 9 × 5 = 45 inducers are presented in the enhanced CIT, and 45 out of the 90 other items are preceded by an inducer.)

The main task is preceded by a comprehension check and two practice tasks. The check serves to ensure that the participant fully understands the task. This phase includes all items presented once, including each of the nine possible inducers for the enhanced version. In this phase, participants has plenty of time (10 s) to choose a response – however, each trial requires a correct response. In case of an incorrect response, the phase has to be repeated.

In the following first practice task, the response window is longer (2 s), while the second practice task had the same design as the main task (1000 ms response window). Both practice tasks consist of 9-14 trials, containing all possible main items in each, and several random inducer items in case of the enhanced and no-target versions. In either practice task, in case of too few valid responses, the participants receives a corresponding feedback, and have to repeat the practice task. The requirement is a minimum 60% valid responses (correct key between 150 and 1000 ms) for each of the following item types (when the given type exists in the given CIT version): targets; familiarity-referring inducers; unfamiliarity-referring inducers; main items (probes and irrelevants together).

In each block, after the instuctions, the trials can be started pressing space. During the comprehension test and the first practice task, reminding captions are displayed regarding the two reponse keys and their corresponding categories. (Starting from the second practice block (and throughout the entire task), no captions are displayed anymore.)



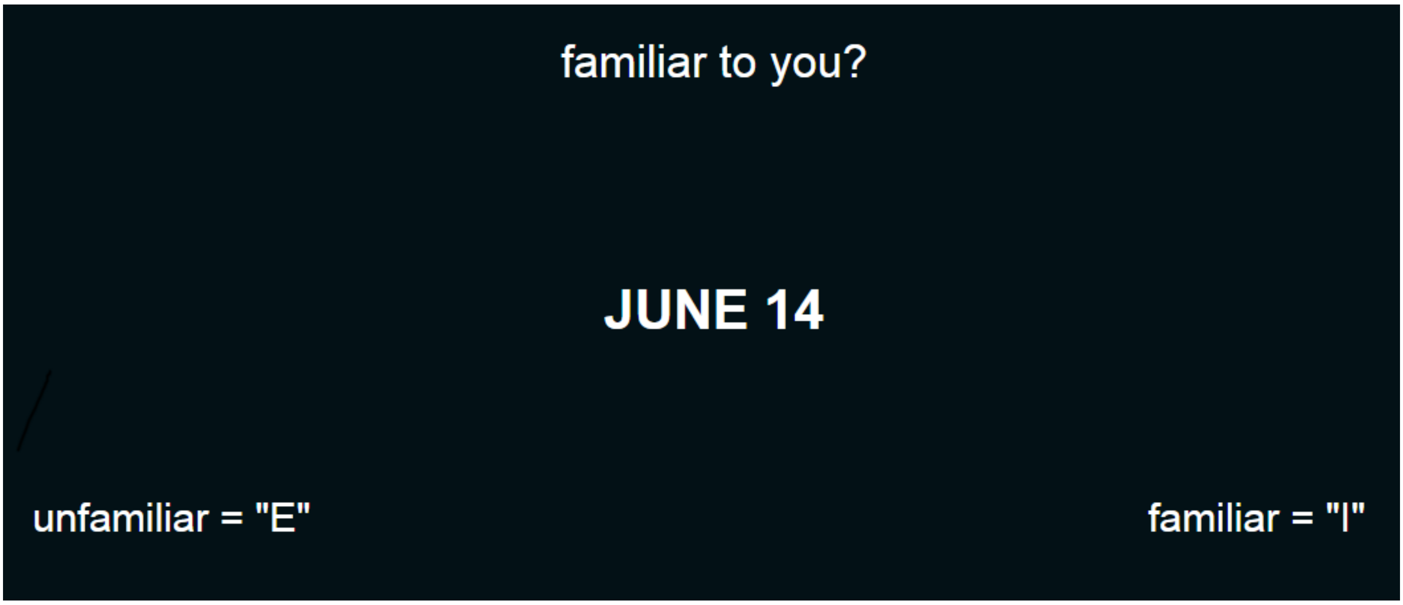
familiar to you?

unfamiliar = "E"

familiar = "I"

Press the space bar to begin.

After pressing space, the task items begin to appear in the middle of the screen.



familiar to you?

JUNE 14

unfamiliar = "E"

familiar = "I"

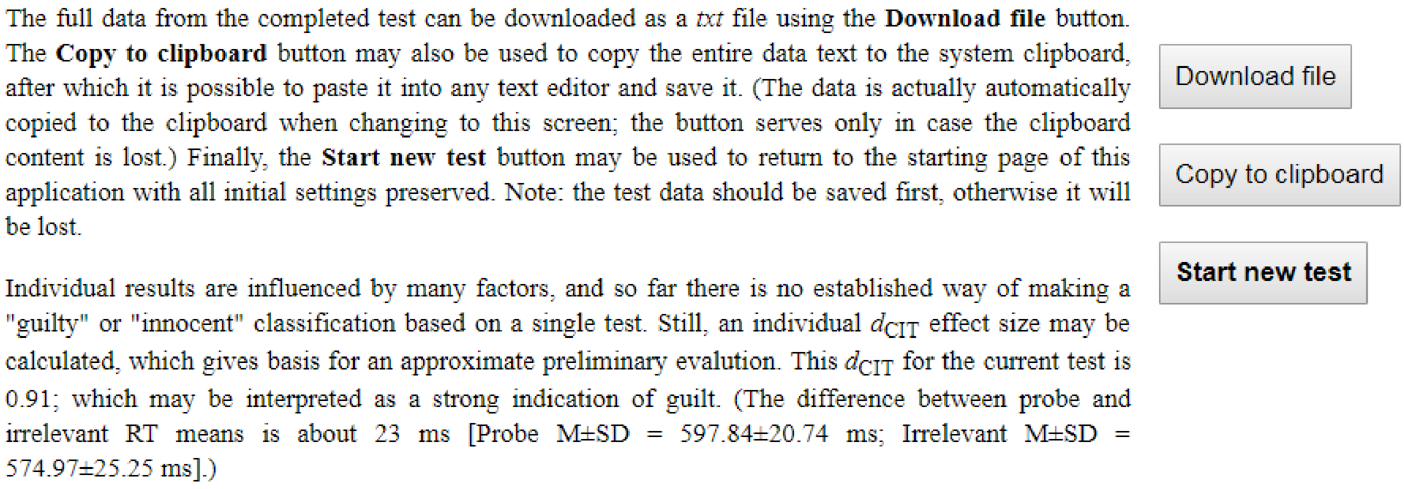
Settings

Currently the settings (apart from those on the main page) involve only the possibility to change the captions on the screen. These all contain their default value on start-up, which can be overwritten if needed. In particular, inducers may be modified to the native language of the examinee. The instructions can be either explained or (since precise translation is not of essence in this case) they can be auto-translated (e.g. Google-translate in Chrome).

Results

At the end of the test, first there is only a simple notification that the test has ended. This has to be closed using a red X button at the bottom right. Afterwards, the full results data can be accessed several ways. Using two separate buttons, it may be downloaded as a file, or copied to the system clipboard so that it may be pasted into an editor and saved. (The data is actually automatically copied to the clipboard at the end of the task; the button serves only in case the clipboard content is lost.)

The following screenshot illustrates the results page in case of an examinee who was reacting significantly slower to the probe item relative to irrelevant items.



The full data from the completed test can be downloaded as a *txt* file using the **Download file** button. The **Copy to clipboard** button may also be used to copy the entire data text to the system clipboard, after which it is possible to paste it into any text editor and save it. (The data is actually automatically copied to the clipboard when changing to this screen; the button serves only in case the clipboard content is lost.) Finally, the **Start new test** button may be used to return to the starting page of this application with all initial settings preserved. Note: the test data should be saved first, otherwise it will be lost.

Individual results are influenced by many factors, and so far there is no established way of making a "guilty" or "innocent" classification based on a single test. Still, an individual d_{CIT} effect size may be calculated, which gives basis for an approximate preliminary evalution. This d_{CIT} for the current test is 0.91; which may be interpreted as a strong indication of guilt. (The difference between probe and irrelevant RT means is about 23 ms [Probe M±SD = 597.84±20.74 ms; Irrelevant M±SD = 574.97±25.25 ms].)

The data contains all responses (one per row) from the test, with all relevant information, corresponding to the self-explanatory titles in the first row.

subject_id*: The subject identification text as given on the starting page.*

cit_version*: The CIT version as selected on the starting page.*

block_number*: The number of the block.*

trial_number*: The number of the trial.*

stimulus_shown*: The specific text of the stimulus that was displayed on the screen.*

category*: Either "main_item" (probe, irrelevant, or target), or "inducer" (familiar-referring or unfamiliar-referring).*

stim_type*: More specific type: probe, irrelevant, target, familiar-referring inducer, or unfamiliar-referring inducer.*

response_key*: The response key that was used ("e" or "i"), regardless whether it was correct or not.*

rt*: The response time (from the start of the presentation of the stimulus until the pressing of the response key); in ms.*

incorrect*: Whether the response was correct: 1 for incorrect, 0 for correct.*

too_slow*: Whether the response was too slow: 1 for too slow, 0 for not too slow.*

date_in_ms*: The number of milliseconds elapsed since January 1, 1970, 00:00:00 UTC. (This documents the exact time and duration of the experiment, allows to calculate the length of pauses taken between the blocks, etc.)*

Evaluation

As an intra-individual effect size (*Noordraven & Verschuere, 2013*), a d_{CIT} can be calculated as the probe RT mean minus the irrelevant RT mean, divided by the standard deviation of irrelevant RTs: ($M_{\text{RT(probe)}} - M_{\text{RT(irrelevant)}}$) / $SD_{\text{RT(irrelevant)}}$

This can be calculated from the results file, but it is also already automatically calculated and displayed at the end of each test in this application.

The values of this measure typically fall between around -0.3 and 0.8; where a larger number always indicates a larger likelihood of guilt (i.e., recognition of the probe). The optimal cut-off value used as a boundary for the evaluation as guilty or innocent varies by study.

Still, for an immediate informative feedback for each test in this application, evaluative labels were implemented with boundaries based on a rough approximation in view of previous results. These boundaries are depicted in the following table.

Boundaries	Evaluative label
d_{CIT} > 0.4	strong indication of guilt
d_{CIT} > 0.3 and d_{CIT} <=0 .4	fair indication of guilt
d_{CIT} > 0.1 and d_{CIT} <=0 .3	weak indication of guilt
d_{CIT} > 0 and d_{CIT} <=0 .1	indeterminate
d_{CIT} > -0.1 and d_{CIT} <=0	weak indication of innocence
d_{CIT} <=- 0.1	fair indication of innocence

(Note that there is no strong indication of innocence: a very fast probe response is no more expected in case of innocence than in case of guilt.)

Testing and example usage

To experimentally test this application, one may for example enter the personal name (e.g., forename) of a participant as probe, and other, similar names as target and irrelevant items. (These latter items should be controlled for similarity to the extent possible; e.g., number of characters or syllables, word frequency, etc.) In this case, the participant's responses to the probe (own personal name) should be notably slower than the responses to the irrelevants (e.g., around 15-30 ms average difference). Correspondingly, the d_{CIT} should be relatively high as well (e.g., larger than 0.3).

As a control condition, one may enter irrelevant names for all item types, so that the probe, irrelevant, and target items are all in fact irrelevant to the participant. In this case, the responses to the probe should be of similar speed as the responses to the irrelevants (since, again, the probe is not relevant to the participant and is not recognized as any different from the rest of the items).

However, there is also a possibility to programmatically simulate human responses. This is done by a designated testing function that simulates human keypresses whenever an item in the task is displayed. This simulation can be activated by simply entering the function *citapp_testing_on()* into the Chrome DevTools Console (F12) anytime. By default, this function gives 100% correct responses in the first practice round (where this is required), and, for the rest of the task, 90% correct and 10% incorrect responses, with artificial response time randomly ranging from 350 to 600. These numbers can be changed by modifying any of the following parameters: *apptest_correct_ratio_first* (ratio from 0.0 to 1.0), *apptest_correct_ratio* (ratio from 0.0 to 1.0), *apptest_simulated_rt_min* (number in ms), and *apptest_simulated_rt_max* (number in ms). For example, entering *apptest_correct_ratio_first = 0.95, apptest_correct_ratio = 0.8, apptest_simulated_rt_min = 200, apptest_simulated_rt_max = 1050* into the Console will simulate a participant who has 95% correct responses in the first practice round, and, for the rest of the task, 80% correct and 20% incorrect responses, with artificial response time randomly ranging from 200 to 1050 (hence, some of them over the response time limit of 1000, resulting in "too slow" responses). The simulation automatically proceeds to the next block (within a few seconds) as long as there are blocks left in the task. This can be changed using the parameter *apptest_move_to_next_block* (true or false), e.g. *apptest_move_to_next_block = false* will disable this, and the user has to press the appropriate button to start each block. By default, all items have the same simulated responses. However, guilt may be simulated using the variable *apptest_probe_delay* (number in ms), that adds a specified delay whenever the response is given to the probe. E.g., *apptest_probe_delay = 30* will always add 30 ms to the probe response. To stop testing (i.e., stop simulating keypresses), enter *citapp_testing_off()* anytime.

Two example output files are added under the *output_examples* folder. One is the result of simulating a guilty suspect, with settings: *apptest_correct_ratio_first = 1, apptest_correct_ratio = 0.95, apptest_simulated_rt_min = 500, apptest_simulated_rt_max = 650, apptest_probe_delay = 25*. The resulting d_{CIT} was 0.58, and the raw difference between probe and irrelevant RT means was 24 ms. The file name is *CIT_app_test_sim_guilty_standard_2block_20181120002430.txt*. The other is the result of simulating an innocent examinee, with default settings (*apptest_correct_ratio_first = 1, apptest_correct_ratio = 0.9, apptest_simulated_rt_min = 350, apptest_simulated_rt_max = 600,* *apptest_probe_delay = 0*). The resultingd_{CIT} was 0.03, and the raw difference between probe and irrelevant RT means was 2 ms. The file name is *CIT_app_test_sim_innocent_enhanced_3block_20181120003753.txt*.

For convenience, there is a *Demo* menu, under which the start page may be automatically filled by a simple button click. (The description of the example is given under the same menu.)

Finally, for developement purposes, there is a *dev_test()* function, which executes all main functions that initiate the test, prepare the conditions, and, most importantly, generate the test items. The start page input fields are filled automatically using the *Demo* data and settings. The function also checks whether, in case of the standard and enhanced CIT versions, the expected number of items are generated with the expected item texts (i.e., with the words that are entered on the start page). If no error occurs, the approval message is logged in the Console. Additionally, the function may be given a single argument: the number 1, the number 2, or the number 3. Entering any of these three will automatically initiate human simulation (with *apptest_probe_delay* set to 25) after (and if) the previously described test was passed. Choosing *dev_test(1)* will complete the standard version, *dev_test(2)* will complete the enhanced version, and *dev_test(3)* will complete the no-target version. In case of no issues, the test should complete itself automatically, all the way until the end of test page.

This *dev_test()* function automatically runs via Travis CI on each *git push*.
(Link: https://travis-ci.org/gasparl/citapp_pc)

Planned features

- More settings; in particular the possibility to change any of the instruction texts
- Graphic (pictorial) stimuli

I would particularly appreciate contributions regarding:

- Possibility to store settings locally
- The automated (Travis/karma) testing should load the CITapp.html from the root (now it's implemented with a "dummy" HTML insertion)
- More detailed end feedback; aggregated data per item type; graphs based on the results (e.g. RT density per item)
- Translations to other languages (including their implementation in HTML)

Support

If you have any questions or find any issues (bugs, desired features), write an email or open a new issue.

Citation

Cite this application as:

Lukács, G. (2019). CITapp - a response time-based Concealed Information Test lie detector web application. *Journal of Open Source Software, 4*(34), 1179, https://doi.org/10.21105/joss.01179

References

Lukács, G. & Ansorge, U. (2019). Information leakage in the response time-based Concealed Information Test. *Applied Cognitive Psychology*. https://doi.org/10.1002/acp.3565

Lukács, G., Kleinberg, B., & Verschuere, B. (2017). Familiarity-related fillers improve the validity of reaction time-based memory detection. *Journal of Applied Research in Memory and Cognition, 6*(3), 295–305. https://doi.org/10.1016/j.jarmac.2017.01.013

Noordraven, E., & Verschuere, B. (2013). Predicting the Sensitivity of the Reaction Time-based Concealed Information Test: Detecting deception with the Concealed Information Test. *Applied Cognitive Psychology, 27*(3), 328–335. https://doi.org/10.1002/acp.2910

Seymour, T. L., Seifert, C. M., Shafto, M. G., & Mosmann, A. L. (2000). Using response time measures to assess “guilty knowledge”. *Journal of Applied Psychology, 85*(1), 30–37. https://doi.org/10.1037//0021-9010.85.1.30