03 CSI online typing: Manual answer classification and participant exclusion

Kirsten Stark

01 November, 2021

Load packages

```
rm(list = ls())

library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Load data and answercodes

```
options( "encoding" = "UTF-8" )

# input
input = "data_long.csv"

# load data
df <- read.csv(here::here("data", "transient_data_files", input))

# load answer codes
answercodes <- read.csv(here::here("data", "supplementary_info", "answercodes.csv"),sep = ";")
```

Check and classify participants' answer behavior

Assign answer codes to participants' typed naming

```
df$answercode <- NA
for(i in 1:nrow(answercodes)) {
  df$answercode[toupper(df$item) == toupper(answercodes$item[i]) &
                  toupper(df$word) == toupper(answercodes$word[i])] <-
    answercodes$answercode[i]
}
df$answercode <- ifelse(is.na(df$word), "isna", df$answercode)
```

Have all answers been classified?

```
sum(is.na(df$answercode)) == 0
```

```
## [1] TRUE
```

Overview of answer codes (naming mistakes) - BEFORE participant exclusion:

```
table(df$answercode)
```

```
##
##          almostcorrect  backspace_space_enter                correct
##                    861                     34                  3762
## first_letter_incorrect                   isna      semantic_relation
##                    217                    228                   132
##            shift_start        unrelated_other
##                     13                     33
```

- *Almost correct* naming are either synonyms or naming with only minor or corrected typos where the word is still recognizable and the initial letter is correct.
- Naming was classified as *backspace_space_enter* when the they started by typing the backspace, space, enter, or caps lock key, and then typed some word.
- *Correct* are all word entrys that are identical to the naming participants were familiarized with.
- *First_letter_incorrect* are words/synonymes that are later typed (almost) correct, but where the first letter was wrong, even if it was later on corrected. The entries suggest that participants sometimes continued typing the last word if they hadn't finished to do so.
- *isna* when no typed entry was recorded.
- *semantic_relation* when another semantically related word was typed that was not a synonym.
- *shift_start* when participants started typing the word with pressing the shift button.
- *unrelated_other* are either completely unrelated words, nonwords, single letter entries, or only keys like backspace, space, enter.

For the main analysis, we consider correct and almost correct entries as correct:

```
df$correct <- NA
df$correct <- ifelse(df$answercode == "correct" | df$answercode == "almostcorrect", 1, NA)

sum(df$answercode == "correct") + sum(df$answercode == "almostcorrect") ==
  sum(df$correct, na.rm = TRUE)
```

```
## [1] TRUE
```

How many trial onset times are considered as invalid?

```
# Amount and percentage of NA or excluded trials
(sumna <- sum(is.na(df$correct)))
```

```
## [1] 657
```

```
(percentagena <- sum(is.na(df$correct))/nrow(df))
```

```
## [1] 0.1244318
```

In a total of 657 trials participants did not enter anything or entered a chunks of characters that were not considered as correct based on the preregistered criteria. This is 0.12 % of all trials.

Trials with missing values per participant:

```
for (i in 1:length(unique(df$type))) {
  print(paste(unique(df$type)[i],"data collection: ", sep = " "))
  print("Amount of trials without any entry for timing.01")
  print(as.data.frame(table(
    df$subject[df$type == unique(df$type)[i]],
    is.na(df$timing.01[df$type == unique(df$type)[i]]))) %>%
      filter(Var2 == TRUE) %>%
  dplyr::rename(subject = Var1, totaltrials = Var2, NA_trials = Freq) %>%
  mutate(totaltrials = 160) %>%
    mutate(percentage_NA = NA_trials/totaltrials))
}
```

```
## [1] "main data collection: "
## [1] "Amount of trials without any entry for timing.01"
##    subject totaltrials NA_trials percentage_NA
## 1        1         160         2       0.01250
## 2        2         160         2       0.01250
## 3        3         160         7       0.04375
## 4        4         160         1       0.00625
## 5        5         160         4       0.02500
## 6        6         160         7       0.04375
## 7        7         160         2       0.01250
## 8        8         160         2       0.01250
## 9        9         160        51       0.31875
## 10      10         160         2       0.01250
## 11      11         160         1       0.00625
## 12      12         160         2       0.01250
## 13      13         160         4       0.02500
## 14      14         160         4       0.02500
## 15      15         160         2       0.01250
## 16      16         160         6       0.03750
## 17      17         160         0       0.00000
## 18      18         160         4       0.02500
```

```
## 19      19         160         2       0.01250
## 20      20         160         5       0.03125
## 21      21         160         2       0.01250
## 22      22         160         5       0.03125
## 23      23         160         1       0.00625
## 24      24         160         3       0.01875
## 25      25         160         8       0.05000
## 26      26         160         7       0.04375
## 27      27         160         4       0.02500
## 28      28         160         2       0.01250
## 29      29         160         6       0.03750
## 30      30         160         1       0.00625
## [1] "replacement data collection: "
## [1] "Amount of trials without any entry for timing.01"
##   subject totaltrials NA_trials percentage_NA
## 1       1         160         1       0.00625
## 2       2         160         5       0.03125
## 3       3         160         1       0.00625
```

Amount of trials classified as correct per participant:

```
for (i in 1:length(unique(df$type))) {
  print(paste(unique(df$type)[i],"data collection: ", sep = " "))
  print("Amount of trials classified as correct
        (correct, correct with typos, synonymes")
  print(as.data.frame(table(df$subject[df$type == unique(df$type)[i]],
                      !is.na(df$correct[df$type == unique(df$type)[i]]))) %>%
    filter(Var2 == TRUE) %>%
    dplyr::rename(subject = Var1, totaltrials = Var2, correct = Freq) %>%
      mutate(totaltrials = 160) %>%
      mutate(percentagecorrect = correct/totaltrials))
}
```

```
## [1] "main data collection: "
## [1] "Amount of trials classified as correct \n        (correct, correct with typos, synonymes"
##    subject totaltrials correct percentagecorrect
## 1        1         160     155           0.96875
## 2        2         160     151           0.94375
## 3        3         160     124           0.77500
## 4        4         160     144           0.90000
## 5        5         160     141           0.88125
## 6        6         160     141           0.88125
## 7        7         160     153           0.95625
## 8        8         160     133           0.83125
## 9        9         160      66           0.41250
## 10      10         160     138           0.86250
## 11      11         160     148           0.92500
## 12      12         160     152           0.95000
## 13      13         160     129           0.80625
## 14      14         160     135           0.84375
## 15      15         160     153           0.95625
## 16      16         160     146           0.91250
## 17      17         160     153           0.95625
```

```
## 18      18         160     143           0.89375
## 19      19         160     144           0.90000
## 20      20         160     141           0.88125
## 21      21         160     139           0.86875
## 22      22         160     141           0.88125
## 23      23         160     150           0.93750
## 24      24         160     146           0.91250
## 25      25         160     140           0.87500
## 26      26         160     136           0.85000
## 27      27         160     145           0.90625
## 28      28         160     139           0.86875
## 29      29         160     131           0.81875
## 30      30         160     149           0.93125
## [1] "replacement data collection: "
## [1] "Amount of trials classified as correct \n        (correct, correct with typos, synonymes"
##   subject totaltrials correct percentagecorrect
## 1       1         160     128           0.80000
## 2       2         160     140           0.87500
## 3       3         160     149           0.93125
```

Participant exclusion

Naming performance

As preregistered, participants with an accuracy below 80 % (either because of inaccurate naming or because of no naming) will be excluded and replaced for the main analyses:

```
for (i in 1:length(unique(df$type))) {
  print(paste(unique(df$type)[i],"data collection: ", sep = " "))
   print("Participants with performance below 80 %")
   print(as.data.frame(table(df$subject[df$type == unique(df$type)[i]],
                      !is.na(df$correct[df$type == unique(df$type)[i]]))) %>%
    filter(Var2 == TRUE) %>%
    dplyr::rename(subject = Var1, totaltrials = Var2, correct = Freq) %>%
     mutate(totaltrials = 160) %>%
     mutate(percentagecorrect = correct/totaltrials) %>%
     filter(percentagecorrect < .80))
}
```

```
## [1] "main data collection: "
## [1] "Participants with performance below 80 %"
##   subject totaltrials correct percentagecorrect
## 1       3         160     124            0.7750
## 2       9         160      66            0.4125
## [1] "replacement data collection: "
## [1] "Participants with performance below 80 %"
## [1] subject           totaltrials       correct           percentagecorrect
## <0 rows> (or 0-length row.names)
```

In the main data collection, participants with subject no. 3 and 9 will be excluded from the data analyses.

Comments

```
#table(unique(df$comments[df$comments != "NA"]))
as.data.frame(table(df$type, df$subject, df$comments)) %>% filter(Freq != 0)
```

```
##           Var1 Var2
## 1         main   13
## 2         main    5
## 3         main   19
## 4         main   20
## 5  replacement    2
## 6         main   30
## 7         main   16
## 8         main   25
## 9         main   14
## 10        main   21
##
## 1
## 2                                                                                Beim Benennen der
## 3
## 4                                  Die einzige Anmerkung, die vielleicht hilfreich wäre, ist dass ich
## 5  Ich habe ein wirklich miserables Kurzzeitgedächnis und wusste bei Gegenständen mit mehreren synony
## 6
## 7
## 8
## 9
## 10
##    Freq
## 1   160
## 2   160
## 3   160
## 4   160
## 5   160
## 6   160
## 7   160
## 8   160
## 9   160
## 10  160
```

We will exclude one participant, the participant with the subject no. 20, who indicated that she has been living in England for several years and is writing usually in English and therefore at first only remembered the English word of the items and is unfamiliar with typing Umlaute. Her answer behavior also indicated that she typed the English words several times.

```
unique(df$comments[df$subject == 20 & df$type == "main"])
```

```
## [1] "Die einzige Anmerkung, die vielleicht hilfreich wäre, ist dass ich seit einigen Jahren in Engla
```

Attention checks

1) Item vs. non-item
We had two item- vs. non-item attention checks

```
## Item vs. non-item
# CH01_01 (Taube), CH01_02 (Apfel), CH02_01 (Luftballon) and CH02_02 (Biene) are items and 2 should app
# CH02_03 (Radio), CH02_04 (Sparschwein), CH02_03 (Laptop) and CH02_04 (Wattestäbchen) are non-items an
# --> As subjects had to select two items,
# controlling the selected items is enough
for(i in 1:length(unique(df$type))) {
  print(as.data.frame(df %>% filter(type == unique(df$type)[i]) %>%
  select(type, subject, CH01_01, CH01_02, CH02_01, CH02_02)) %>%
  filter( CH01_01 == 1 | CH01_02 == 1 | CH02_01 == 1 | CH02_02 == 1)%>%
  unique())
}
```

```
##     type subject CH01_01 CH01_02 CH02_01 CH02_02
## 1   main       6       1       2       2       2
## 161 main      11       2       1       2       2
## 321 main      14       2       1       2       2
## 481 main      22       1       2       2       2
## 641 main      24       1       2       2       2
## 801 main      29       1       2       2       2
## [1] type    subject CH01_01 CH01_02 CH02_01 CH02_02
## <0 rows> (or 0-length row.names)
```

6 participants of the main data collection (subject no. 6, 11, 14, 22, 24, 29) made one mistake in the item vs. non-item task after the familiarization phase. However, all performed perfectly in the item vs. non-item task after the main task.

2) Cheating

```
## Did participants cheat
# CH03 = 1 - yes, I worked through it till the end,
# CH03 = 2 - no, I stopped or cheated midway
# CH03 = -9 - no answer
for(i in 1:length(unique(df$type))) {
  print(as.data.frame(df %>% filter(type == unique(df$type)[i]) %>%
  select(type, subject, CH03)) %>%
  filter(CH03 != 1)%>%
  unique())
}
```

```
##   type subject CH03
## 1 main      19    2
## [1] type    subject CH03
## <0 rows> (or 0-length row.names)
```

One participant in the main data collection indicated that they stopped working on the task or cheated midway. However, in the comments, that participant indicated that they only indicated this because they had to open the door during the familiarization phase and therefore stopped for about 2 min. As there was no timeout in the familiarization, this shouldn't have affected the data much. Therefore, we do not exclude this participant.

```
unique(df$comments[df$subject == 19 & df$type == "main"])
```

```
## [1] "Der Grund, weshalb ich angegeben habe, dass ich das Experiment nicht zu Ende bearbeitet habe, is
```

3) Mother tongue

```
table(df$language)
```

```
##
##    1
## 5280
```

```
table(df$language.test)
```

```
##
##    3
## 5280
```

4) Keyboard

```
table(df$keyboard_type)
```

```
##
##    2
## 5280
```

```
table(df$KB02_01)
```

```
##
## ägyptisch
##      5280
```

```
table(df$KB03_01)
```

```
##
## Quote
##  5280
```

```
table(df$KB03_02)
```

```
##
## KeyZ
## 5280
```

```
table(df$KB03_03)
```

```
##
## KeyP
## 5280
```

The keyboard screening worked well for all participants.

Exclude participants and export final data frame

```r
df2 <- df %>%
  filter(!((df$subject == 3 | df$subject == 9 |
              df$subject == 20) & df$type == "main"))# %>%

df2$subject <-  ifelse(df2$subject == 1 & df2$type == "replacement", 9, df2$subject)
df2$subject <- ifelse(df2$subject == 2 & df2$type == "replacement", 20, df2$subject)
df2$subject <- ifelse(df2$subject == 3 & df2$type == "replacement", 3, df2$subject)
table(df2$subject)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
## 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160
##  21  22  23  24  25  26  27  28  29  30
## 160 160 160 160 160 160 160 160 160 160
```

```r
write.csv(df2, here::here("data", "transient_data_files", "data_long_final.csv"))
```