German translation of the Artificial-Social-Agent questionnaire instrument for evaluating human-agent interaction

Transformation from raw data to the input files - second translation round

Boleslav Khodakov

31 May, 2023

Contents

1 Introduction

In this document, we transform a raw data file into an input data file. Because of privacy reasons we did not release for public access the raw data file.

There is one document to be transformed - the English items whose translations previously had low ICC values (below 0.6), with new translations. **Please note: The descriptions provided in this document relate to the dataset containing all participants' data. However, there is also a subset of 29 participants who recommended using their data. This subset is evaluated in the same way until (including) individual ICC values are calculated. After that, we only continue with the complete (30 participant) data.**

We use the following packages:

```
library(haven)    # Use read_sav function
library(dplyr)    # Use select function
library(knitr)    # Get markdown file
library(tinytex)  # Use TeX environment
library(rticles)  # Use CTeX documents template
```

```
library(pander) # For pandering tables
panderOptions("table.alignment.default","left")
```

2 Data file Final_ASA_German_Round_2_anonym.sav

We read the raw data file consisting of the 35 English ASA item scores and corresponding German translation scores. The raw scores are from a 7-point scale, ranging from -3 (disagree), 0 (neither agree nor disagree) to 3 (agree). Before we could do any analysis, we first had to "clean up" the score of items by removing any characters other than numbers (except the minus sign for negative numbers).

```
# Read the file
# This will be the dataframe with all participants
d_ASA_1 <- read_sav("Final_ASA_German_Round_2_anonym.sav")

# Only take into account people who recommend using their data for scientific purposes
# This will be the dataframe with only those participants who recommend
# using their data
d_ASA_2 <- d_ASA_1[d_ASA_1$Use_Data == 'Yes',]

# Cleanup labels (delete the actual words, only numbers remain)
clean_labels <- function(data){
  data[data=="Neither agree nor disagree\n\n0"] <- "0"
  data[data=="Disagree\n\n-3"] <- "-3"
  data[data=="Agree\n\n3"] <- "3"
  data[data=="Weder noch\n\n0"] <- "0"
  data[data=="Stimme nicht zu\n\n-3"] <- "-3"
  data[data=="Stimme zu\n\n3"] <- "3"
  data[data==" \n\n1"] <- "1"
  data[data==" \n\n2"] <- "2"
  data[data==" \n\n-1"] <- "-1"
  data[data==" \n\n-2"] <- "-2"
  return(data)
}

# Clean up labels
d_ASA_1 <- clean_labels(d_ASA_1)
d_ASA_2 <- clean_labels(d_ASA_2)

# Make columns of question items into numeric types (previously was string)
```

```r
d_ASA_1 <- d_ASA_1 %>% mutate(across("AC_E_1":"Q_DE_UAI4_3", as.numeric))
d_ASA_2 <- d_ASA_2 %>% mutate(across("AC_E_1":"Q_DE_UAI4_3", as.numeric))
```

3 Transformation results as input data file for further analysis

3.1 File transformed_data_round_2.sav

We removed scores of attention control questions and other irrelevant data ('Finished' and 'rand_id'), retaining ratings of English items and corresponding German translations. In case some of participants didn't complete the questionnaire survey, the resulting null values were omitted by omitting data rows of these participants all-together. The code excluding people failing attention checks (from the Chinese translation codebase) is still in place. However, for the current study we already excluded participants failing the checks on Qualtrics. **Please note: Since these participants are not paid for their submission, we cannot use or publish their data in any way.**

```r
# Select only question items from dataframe (incl. attention checks)
# dd1 = all participants
dd1 <- data.frame(select(d_ASA_1, AC_E_1:Q_DE_UAI4_3))
# dd2 = "recommended" participants
dd2 <- data.frame(select(d_ASA_2, AC_E_1:Q_DE_UAI4_3))

# Filter out participant entries based on failed attention checks
attention_check_filtering <- function(data){
  # Select attention check questions
  # Note: for German translation rounds,
  # no single row should be removed based on failed attention checks.
  # This is already done in Qualtrics.
  # Note: This code chunk remains for 1) the final evaluation 2) legacy compatibility
  i <- grep("AC",colnames(data))

  # Select desired answers for attention check questions
  Atten <- c(-3,3,3,-3,0,3,-3,0,-3,3,2,3,-3,1)


  # The following code and comments still remain from the Chinese translation code
  # with minor adjustments:

  x <- NULL # Row number of participant who failed the attention check
  for (j in (1:nrow(data))){
  # Find participants who failed attention check in 'dd1'
```

```
    count <- 0
    # The number of incorrectly answered attention control questions of each participant
    for (k in 1:14){

        if (as.numeric(data[[i[k]]][j])!=Atten[k]) # Check whether each participant's

        # attention control question answers are consistent with the correct answers
          count <- count+1

    }
        if (count>2)
        # Row number of the participant who failed more than two
        # attention control questions were added to 'x'
        x <- append(x,j)
        # Participants who failed more than two attention control questions
  }
  m <- length(x) # The number of participants who failed attention check
  if (m!=0)
    data <- data[-x,] # Participants who failed attention check were excluded

  # Print whether any particpants failed attention checks
  cat("Amount of participants who failed attention checks: ", m,
      "\nAmount of participants left after filtering based on attention checks: ",
      nrow(data), "\n")
  return(data)
}

#Perform filtering on "all" and "recommended" dataframes
dd1 <- attention_check_filtering(dd1)
```

```
## Amount of participants who failed attention checks:  0
## Amount of participants left after filtering based on attention checks:  30
```

```
dd2 <- attention_check_filtering(dd2)
```

```
## Amount of participants who failed attention checks:  0
## Amount of participants left after filtering based on attention checks:  29
```

All participants' evaluation data were included as none of the participants failed any of the attention control questions (logically). For the formative translations, we did not yet reverse scores of questionnaire items and German translations with R indication. This can be omitted during the translation

step however. The code is still there for the summative evaluation, where it will be used. Thus, we obtained the output data `transformed_data_round_2.sav` for further evaluation.

```
# Select scores of 44 English items and corresponding German translations
# d1 = all participants
d1 <- as.data.frame(select(dd1, Q_E_HLA4:Q_E_UAI4, Q_DE_HLA4:Q_DE_UAI4_3))
# d2 = "recommended" participants
d2 <- as.data.frame(select(dd2, Q_E_HLA4:Q_E_UAI4, Q_DE_HLA4:Q_DE_UAI4_3))

# For English items which have multiple German translations,
# duplicate the English items x times (x = amount of extra German translations)
duplicate <- function(data){
  # Q_DE_HLA4 (2)
  data <- data %>%
    mutate(Q_E_HLA4_2 = Q_E_HLA4, .after = "Q_E_HLA4")
  # Q_DE_NA1 (2)
  data <- data %>%
    mutate(Q_E_NA1_2 = Q_E_NA1, .after = "Q_E_NA1")
  # Q_DE_NA2 (2)
  data <- data %>%
    mutate(Q_E_NA2_2 = Q_E_NA2, .after = "Q_E_NA2")
  # Q_DE_NA4 (2)
  data <- data %>%
    mutate(Q_E_NA4_2 = Q_E_NA4, .after = "Q_E_NA4")
  # Q_DE_NA5 (2)
  data <- data %>%
    mutate(Q_E_NA5_2 = Q_E_NA5, .after = "Q_E_NA5")
  # Q_DE_NB1 (2)
  data <- data %>%
    mutate(Q_E_NB1_2 = Q_E_NB1, .after = "Q_E_NB1")
  # Q_DE_AAS1 (2)
  data <- data %>%
    mutate(Q_E_AAS1_2 = Q_E_AAS1, .after = "Q_E_AAS1")
  # Q_DE_APP1 (2)
  data <- data %>%
    mutate(Q_E_APP1_2 = Q_E_APP1, .after = "Q_E_APP1")
  # Q_DE_APP2 (2)
  data <- data %>%
    mutate(Q_E_APP2_2 = Q_E_APP2, .after = "Q_E_APP2")
  # Q_DE_UAA3 (2)
```

```r
data <- data %>%
  mutate(Q_E_UAA3_2 = Q_E_UAA3, .after = "Q_E_UAA3")
# Q_DE_AE4 (3)
data <- data %>%
  mutate(Q_E_AE4_2 = Q_E_AE4, .after = "Q_E_AE4")
data <- data %>%
  mutate(Q_E_AE4_3 = Q_E_AE4, .after = "Q_E_AE4_2")
# Q_DE_UE1 (2)
data <- data %>%
  mutate(Q_E_UE1_2 = Q_E_UE1, .after = "Q_E_UE1")
# Q_DE_UE2 (2)
data <- data %>%
  mutate(Q_E_UE2_2 = Q_E_UE2, .after = "Q_E_UE2")
# Q_DE_UE3 (2)
data <- data %>%
  mutate(Q_E_UE3_2 = Q_E_UE3, .after = "Q_E_UE3")
# Q_DE_UAL2 (2)
data <- data %>%
  mutate(Q_E_UAL2_2 = Q_E_UAL2, .after = "Q_E_UAL2")
# Q_DE_UAL3 (2)
data <- data %>%
  mutate(Q_E_UAL3_2 = Q_E_UAL3, .after = "Q_E_UAL3")
# Q_DE_UAL5 (2)
data <- data %>%
  mutate(Q_E_UAL5_2 = Q_E_UAL5, .after = "Q_E_UAL5")
# Q_DE_AA1 (3)
data <- data %>%
  mutate(Q_E_AA1_2 = Q_E_AA1, .after = "Q_E_AA1")
data <- data %>%
  mutate(Q_E_AA1_3 = Q_E_AA1, .after = "Q_E_AA1_2")
# Q_DE_AI1 (3)
data <- data %>%
  mutate(Q_E_AI1_2 = Q_E_AI1, .after = "Q_E_AI1")
data <- data %>%
  mutate(Q_E_AI1_3 = Q_E_AI1, .after = "Q_E_AI1_2")
# Q_DE_AI3 (2)
data <- data %>%
  mutate(Q_E_AI3_2 = Q_E_AI3, .after = "Q_E_AI3")
# Q_DE_AI4 (3)
```

```r
data <- data %>%
  mutate(Q_E_AI4_2 = Q_E_AI4, .after = "Q_E_AI4")
data <- data %>%
  mutate(Q_E_AI4_3 = Q_E_AI4, .after = "Q_E_AI4_2")
# Q_DE_AT2 (2)
data <- data %>%
  mutate(Q_E_AT2_2 = Q_E_AT2, .after = "Q_E_AT2")
# Q_DE_AT3 (2)
data <- data %>%
  mutate(Q_E_AT3_2 = Q_E_AT3, .after = "Q_E_AT3")
# Q_DE_SP2 (2)
data <- data %>%
  mutate(Q_E_SP2_2 = Q_E_SP2, .after = "Q_E_SP2")
# Q_DE_SP3 (3)
data <- data %>%
  mutate(Q_E_SP3_2 = Q_E_SP3, .after = "Q_E_SP3")
data <- data %>%
  mutate(Q_E_SP3_3 = Q_E_SP3, .after = "Q_E_SP3_2")
# Q_DE_IIS3 (2)
data <- data %>%
  mutate(Q_E_IIS3_2 = Q_E_IIS3, .after = "Q_E_IIS3")
# Q_DE_IIS4 (2)
data <- data %>%
  mutate(Q_E_IIS4_2 = Q_E_IIS4, .after = "Q_E_IIS4")
# Q_DE_AEI1 (2)
data <- data %>%
  mutate(Q_E_AEI1_2 = Q_E_AEI1, .after = "Q_E_AEI1")
# Q_DE_AEI3 (2)
data <- data %>%
  mutate(Q_E_AEI3_2 = Q_E_AEI3, .after = "Q_E_AEI3")
# Q_DE_AEI4 (2)
data <- data %>%
  mutate(Q_E_AEI4_2 = Q_E_AEI4, .after = "Q_E_AEI4")
# Q_DE_AEI5 (3)
data <- data %>%
  mutate(Q_E_AEI5_2 = Q_E_AEI5, .after = "Q_E_AEI5")
data <- data %>%
  mutate(Q_E_AEI5_3 = Q_E_AEI5, .after = "Q_E_AEI5_2")
# Q_DE_UAI1 (3)
```

```
  data <- data %>%
    mutate(Q_E_UAI1_2 = Q_E_UAI1, .after = "Q_E_UAI1")
  data <- data %>%
    mutate(Q_E_UAI1_3 = Q_E_UAI1, .after = "Q_E_UAI1_2")
  # Q_DE_UAI2 (2)
  data <- data %>%
    mutate(Q_E_UAI2_2 = Q_E_UAI2, .after = "Q_E_UAI2")
  # Q_DE_UAI3 (2)
  data <- data %>%
    mutate(Q_E_UAI3_2 = Q_E_UAI3, .after = "Q_E_UAI3")
  # Q_DE_UAI4 (3)
  data <- data %>%
    mutate(Q_E_UAI4_2 = Q_E_UAI4, .after = "Q_E_UAI4")
  data <- data %>%
    mutate(Q_E_UAI4_3 = Q_E_UAI4, .after = "Q_E_UAI4_2")
  return(data)
}



# Perform duplication on both dataframes
d1 <- duplicate(d1)
d2 <- duplicate(d2)

# ASIMO is hardcoded to 0
d1$AgentID <- 0 # Add a column 'AgentID' to facilitate analysis for comparison
d2$AgentID <- 0 # Add a column 'AgentID' to facilitate analysis for comparison
```