

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

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

Boleslav Khodakov

06 June, 2023

Contents

1	Introduction	1
2	Data file Final_ASA_German_Round_3_anonym.sav	2
3	Transformation results as input data file for further analysis	2
3.1	File transformed_data_round_3.sav	2

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.

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(rarticles) # Use CTeX documents template
library(pander)   # For pandering tables
panderOptions("table.alignment.default", "left")
```

2 Data file Final_ASA_German_Round_3_anonym.sav

We read the raw data file consisting of the 13 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
d_ASA_1 <- read_sav("Final_ASA_German_Round_3_anonym.sav")

# Only take into account people who recommend using their data for scientific purposes
#d_ASA_1 <- d_ASA_1[d_ASA_1$Use_Data == 'Yes',]

# Cleanup labels (delete the actual words, only numbers remain)
d_ASA_1[d_ASA_1=="Neither agree nor disagree\n\n0"] <- "0"
d_ASA_1[d_ASA_1=="Disagree\n\n-3"] <- "-3"
d_ASA_1[d_ASA_1=="Agree\n\n3"] <- "3"
d_ASA_1[d_ASA_1=="Weder noch\n\n0"] <- "0"
d_ASA_1[d_ASA_1=="Stimme nicht zu\n\n-3"] <- "-3"
d_ASA_1[d_ASA_1=="Stimme zu\n\n3"] <- "3"
d_ASA_1[d_ASA_1==" \n\n1"] <- "1"
d_ASA_1[d_ASA_1==" \n\n2"] <- "2"
d_ASA_1[d_ASA_1==" \n\n-1"] <- "-1"
d_ASA_1[d_ASA_1==" \n\n-2"] <- "-2"

# Make columns of question items into numeric types (previously was string)
d_ASA_1 <- d_ASA_1 %>% mutate(across("AC_E_1":"Q_DE_AT2_4", as.numeric))
```

3 Transformation results as input data file for further analysis

3.1 File transformed_data_round_3.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.**

```

# Select only question items from dataframe (incl. attention checks)
dd1 <- data.frame(select(d_ASA_1, AC_E_1:Q_DE_AT2_4))

# 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(dd1))

# Select desired answers for attention check questions
Atten <- c(-3,3,3,0,-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(dd1))){
# Find participants who failed attention check in 'dd1'
count <- 0
# The number of incorrectly answered attention control questions of each participant
for (k in 1:10){

  if (as.numeric(dd1[[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)
dd1 <- dd1[-x,] # Participants who failed attention check were excluded

```

```
# Print whether any participants failed attention checks
cat("Amount of participants who failed attention checks: ", m,
    "\nAmount of participants left after filtering based on attention checks: ", nrow(dd1))
```

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

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_3.sav` for further evaluation.

```
# Select scores of 44 English items and corresponding German translations
d1 <- as.data.frame(select(dd1, Q_E_HLA4:Q_E_AT2, Q_DE_HLA4:Q_DE_AT2_4))

# For English items which have multiple German translations,
# duplicate the English items x times (x = amount of extra German translations)

# Q_DE_HLA4 (4)
d1 <- d1 %>%
  mutate(Q_E_HLA4_2 = Q_E_HLA4, .after = "Q_E_HLA4")
d1 <- d1 %>%
  mutate(Q_E_HLA4_3 = Q_E_HLA4, .after = "Q_E_HLA4_2")
d1 <- d1 %>%
  mutate(Q_E_HLA4_4 = Q_E_HLA4, .after = "Q_E_HLA4_3")
# Q_DE_NA1 (4)
d1 <- d1 %>%
  mutate(Q_E_NA1_2 = Q_E_NA1, .after = "Q_E_NA1")
d1 <- d1 %>%
  mutate(Q_E_NA1_3 = Q_E_NA1, .after = "Q_E_NA1_2")
d1 <- d1 %>%
  mutate(Q_E_NA1_4 = Q_E_NA1, .after = "Q_E_NA1_3")
# Q_DE_UEI2 (4)
d1 <- d1 %>%
  mutate(Q_E_UE2_2 = Q_E_UE2, .after = "Q_E_UE2")
d1 <- d1 %>%
  mutate(Q_E_UE2_3 = Q_E_UE2, .after = "Q_E_UE2_2")
d1 <- d1 %>%
```

```

mutate(Q_E_UE2_4 = Q_E_UE2, .after = "Q_E_UE2_3")
# Q_DE_UAL3 (4)
d1 <- d1 %>%
  mutate(Q_E_UAL3_2 = Q_E_UAL3, .after = "Q_E_UAL3")
d1 <- d1 %>%
  mutate(Q_E_UAL3_3 = Q_E_UAL3, .after = "Q_E_UAL3_2")
d1 <- d1 %>%
  mutate(Q_E_UAL3_4 = Q_E_UAL3, .after = "Q_E_UAL3_3")
# Q_DE_UAL5 (4)
d1 <- d1 %>%
  mutate(Q_E_UAL5_2 = Q_E_UAL5, .after = "Q_E_UAL5")
d1 <- d1 %>%
  mutate(Q_E_UAL5_3 = Q_E_UAL5, .after = "Q_E_UAL5_2")
d1 <- d1 %>%
  mutate(Q_E_UAL5_4 = Q_E_UAL5, .after = "Q_E_UAL5_3")
# Q_DE_IIS3 (4)
d1 <- d1 %>%
  mutate(Q_E_IIS3_2 = Q_E_IIS3, .after = "Q_E_IIS3")
d1 <- d1 %>%
  mutate(Q_E_IIS3_3 = Q_E_IIS3, .after = "Q_E_IIS3_2")
d1 <- d1 %>%
  mutate(Q_E_IIS3_4 = Q_E_IIS3, .after = "Q_E_IIS3_3")
# Q_DE_IIS4 (4)
d1 <- d1 %>%
  mutate(Q_E_IIS4_2 = Q_E_IIS4, .after = "Q_E_IIS4")
d1 <- d1 %>%
  mutate(Q_E_IIS4_3 = Q_E_IIS4, .after = "Q_E_IIS4_2")
d1 <- d1 %>%
  mutate(Q_E_IIS4_4 = Q_E_IIS4, .after = "Q_E_IIS4_3")
# Q_DE_AEI3 (4)
d1 <- d1 %>%
  mutate(Q_E_AEI3_2 = Q_E_AEI3, .after = "Q_E_AEI3")
d1 <- d1 %>%
  mutate(Q_E_AEI3_3 = Q_E_AEI3, .after = "Q_E_AEI3_2")
d1 <- d1 %>%
  mutate(Q_E_AEI3_4 = Q_E_AEI3, .after = "Q_E_AEI3_3")
# Q_DE_AEI5 (4)
d1 <- d1 %>%
  mutate(Q_E_AEI5_2 = Q_E_AEI5, .after = "Q_E_AEI5")

```

```

d1 <- d1 %>%
  mutate(Q_E_AEI5_3 = Q_E_AEI5, .after = "Q_E_AEI5_2")
d1 <- d1 %>%
  mutate(Q_E_AEI5_4 = Q_E_AEI5, .after = "Q_E_AEI5_3")
# Q_DE_UAI2 (4)
d1 <- d1 %>%
  mutate(Q_E_UAI2_2 = Q_E_UAI2, .after = "Q_E_UAI2")
d1 <- d1 %>%
  mutate(Q_E_UAI2_3 = Q_E_UAI2, .after = "Q_E_UAI2_2")
d1 <- d1 %>%
  mutate(Q_E_UAI2_4 = Q_E_UAI2, .after = "Q_E_UAI2_3")
# Q_DE_AEI1 (4)
d1 <- d1 %>%
  mutate(Q_E_AEI1_2 = Q_E_AEI1, .after = "Q_E_AEI1")
d1 <- d1 %>%
  mutate(Q_E_AEI1_3 = Q_E_AEI1, .after = "Q_E_AEI1_2")
d1 <- d1 %>%
  mutate(Q_E_AEI1_4 = Q_E_AEI1, .after = "Q_E_AEI1_3")
# Q_DE_UEP2 (4)
d1 <- d1 %>%
  mutate(Q_E_UEP2_2 = Q_E_UEP2, .after = "Q_E_UEP2")
d1 <- d1 %>%
  mutate(Q_E_UEP2_3 = Q_E_UEP2, .after = "Q_E_UEP2_2")
d1 <- d1 %>%
  mutate(Q_E_UEP2_4 = Q_E_UEP2, .after = "Q_E_UEP2_3")
# Q_DE_AT2 (4)
d1 <- d1 %>%
  mutate(Q_E_AT2_2 = Q_E_AT2, .after = "Q_E_AT2")
d1 <- d1 %>%
  mutate(Q_E_AT2_3 = Q_E_AT2, .after = "Q_E_AT2_2")
d1 <- d1 %>%
  mutate(Q_E_AT2_4 = Q_E_AT2, .after = "Q_E_AT2_3")

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

```