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

Underlying Analyses - first translation round

Boleslav Khodakov

24 May, 2023

Contents

1 Introduction

This document presents statistical analyses of correlation and variation between English and German ASA questionnaires for the item level. For the first translation round, the translations of 90 English items are evaluated. There are a total of 100 translations, as some items have multiple alternative translations. These are 'Q_E_HLB5', 'Q_E_AL1', 'Q_E_AS1', 'Q_E_AS2', 'Q_E_UE2', 'Q_E_UT2', 'Q_E_UAL1'. The items behind these labels can be found in the "Legend" file. **Please note: The descriptions provided in this document relate to the dataset containing all participants' data. However, there is also a subset of 57 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 (60 participant) data.**

We use the following packages:

```
library(nlme)    # Run multilevel linear models
library(car)     # Package linear regression
library(haven) # Use read_sav fuction
library(dplyr) # Use select function
```

```r
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 files results_round_1_first_half_transformed.sav and results_round_1_second_half_transformed.sav

The input data used in the analysis were transformed from two raw data files 'Final_ASA_German_Round_1_First_H and 'Final_ASA_German_Round_1_Second_Half_anonym.sav'.The detailed transformation from raw data to the input data files was explained in the markdown file 'Transformation from raw data to the input data files' (either for the first or second half).

We divided the 90 ASA items and participants into two groups to control fatigue effects. In the first group, human-ASA interaction evaluation data of the first 44 items (Construct 1-8: the first 12 constructs/dimensions) were collected from 30 bilingual participants with German mother tongue who are native German and fluent English speakers. In the second group, human-ASA interaction evaluation data of the last 46 items were collected from 30 bilingual participants with German mother tongue who are native German and fluent English speakers. Bilingual participants rated human-ASA interaction on 44/46 English items and corresponding German translations plus 14 attention control questions. All participants' evaluation data were included as they failed no attention control questions. We removed irrelevant data, e.g., attention control questions, just retaining scores of English items and corresponding German translations, also with 'AgentID'. We did not yet invert reverse-scoring questionnaire items and their English translations. The steps above were conducted and explained in the markdown files 'Transformation from raw data to the input data files', resulting in a two data file 'transformed_data_round_1_half_1.sav' and 'transformed_data_round_1_half_2.sav'. Up to this step, rating scores of 44/46 English items and corresponding German translations were ready for further analysis.

```r
# Load data for all participants
data01 <- data.frame(read_sav("transformed_data_round_1_half_1.sav"))
# Load data for "recommended" participants
data01_recommended <- data.frame(read_sav("transformed_data_round_1_half_1_recommended.sav"))
# Select 44 item scores for English and German translation scores
d1 <- select(data01, Q_E_HLA1:Q_DE_R_AE4)
d1_recommended <- select(data01_recommended, Q_E_HLA1:Q_DE_R_AE4)
```

```r
# Load data for all participants
data02 <- data.frame(read_sav("transformed_data_round_1_half_2.sav"))
```

```r
# Load data for "recommended" participants
data02_recommended <- data.frame(read_sav("transformed_data_round_1_half_2_recommended.sav"))
# Select 46 item scores for English and German translation scores
d2 <- select(data02, Q_E_UE1:Q_DE_UAI4)
d2_recommended <- select(data02_recommended, Q_E_UE1:Q_DE_UAI4)
```

3 Analyses results

3.1 Correlation between English and German ASA Questionnaire

We combined the scores of 44 items and 46 items as well as their corresponding translations in dataframes 'd1' and 'd2'. Then we calculated ICC values for the 90 items. The multilevel model that we fit on the data set is a random intercept model. This model includes a fixed intercept (~1) and participant as a random intercept, indicated by random = ~1|id. Here, 'id' indicates the participant code for 30 bilingual participants whose scores were used to calculate ICC values.

3.1.1 ICC values for 100 items

We combined the scores of 44 items and 46 items as well as their corresponding translations in data frames 'd1' and 'd2'. Then we calculated ICC values for the 90 items. Items were duplicated leading to a total of 100 (so each of the multiple translations had one match). The multilevel model that we fit on the data set is a random intercept model. This model includes a fixed intercept (~1) and participant as a random intercept, indicated by random = ~1|id. Here, 'id' indicates the participant code for 30 bilingual participants whose scores were used to calculate ICC values. We calculated ICC as: $\rho_I = \frac{\tau^2}{\tau^2+\sigma^2}$ whereby τ^2 is the variance between participants, and σ^2 is the variance within the score of individual (Finch, Bolin, and Kelley 2019). For the ICC calculation we defined the *getICC* function.

```r
getICC <-function(model)
# Function for ICC value calculation using multilevel linear model
{
  vc.model <- VarCorr(model)
  # Estimated variances and correlations between the random-effects terms
  sigma_var <-as.numeric(vc.model[2,1])
  # Variance within the groups
  tau_var <- as.numeric(vc.model[1,1])
  # Variance between the groups
  icc <- tau_var/(tau_var + sigma_var)
  # Calculate ICC value
  return(icc)
}
```

Data frames 'd1' and 'd2' both have 30 data points, which we combined in single data frame.

```
# Combine evaluation scores of 44 items and 46 items for all participants
d_total <- cbind(select(d1,Q_E_HLA1:Q_E_R_AE4), select(d2,Q_E_UE1:Q_E_UAI4),
           select(d1,Q_DE_HLA1:Q_DE_R_AE4),select(d2,Q_DE_UE1:Q_DE_UAI4))

# We are not able to use `cbind` on the recommended dataframes, as they are asymmetrical. Those fr
```

Next, we defined a function to run a multilevel model and obtain the associated ICC value for that model. As input, this function accepts the scores in both languages and the participant ID number. Before the model can be fitted this input data is transformed into a long format. The function returns ICC in value.

```
getLME <-function(s_1,s_2)
# Function for a linear mixed-effects model
{
  id<-rownames(s_2)
  # Row names that represent the ID number of each participant
  score_German<- data.frame(id, s_1, language= 1)
  # Transform German scores from wide format to long format and label as 1
  Score_English<- data.frame(id, s_2, language= 2)
  # Transform English scores from wide format to long format and label as 2
  Score_total <- rbind(score_German, Score_English)
  # Combine German and English scores in the long format
  m0 <- lme(score ~ 1, data = Score_total, random = ~1|id, method = "ML")
  # Linear mixed-effects model with a fixed intercept and
  # a random intercept of participant's ID number
  return(getICC(m0))
}
```

With the *getLME* function defined, the next step is to use this function to calculate the ICC value for each of the 90 (100, counting duplicated columns for multiple translations) ASA questionnaire items, and in addition, calculate the grand mean of these 100 ICC values. When going to the list of ASAQ items, we use the fact that in the data frame the first 100 columns (incl. duplicates) present the results of the English ASAQ version and the last 100 (unique) columns present the results of the German ASAQ version. **Please note: There is no simple way of displaying only the best (highest-ICC) German items for English items with multiple translations. Below, you will find all translations.**

```
calculate_item_ICC_values <- function(data,
                                      icc_output_caption, n=100){
```

```r
l_ICC <- data.frame(ItemID = double(), Item = character(), icc = double())

# Numbers of columns in d_total
German_column_offset <- ncol(data) /2

#Initialize output file for low-ICC combinations
write(icc_output_caption,file="ICC_output.txt",append=TRUE)


# The value of n is equal to the number of columns divided by 2.
for (i in 1:n)
# Go step by step to 90 (100) items of the ASA questionnaire,
# whereby i is the ASA questionnaire item number
{

  # Select scores of German version of ASAQ item i
  score_German <- data.frame(score=data[,i + German_column_offset])

  # Select scores of English version of ASAQ items i
  score_English <- data.frame(score=data[,i])


  # Calculated ICC and add it to the list of ICC values,
  # with ID number of the ASA questionnaire item
  l_ICC <- rbind(l_ICC, data.frame (i, icc = getLME(score_German, score_English)))

  # Get the current ICC value from l_ICC (which is a table)
  real_ICC <- round(l_ICC[i,2], digits=4)


  # For values which are under the threshold of 'Good' (0.6)
  if(0.6 > real_ICC){

    # Create an entry in a text file, to have a list of dissatisfactory translations
    output1 <- paste("l_ICC", real_ICC, sep=" ")
    output1 <- paste(output1, colnames(data[i]), sep=" ")
    output1 <- paste(output1, colnames(data[i + German_column_offset]), sep=" ")

    # Append a text file (of the user's choice) with the entry
```

```
      write(output1,file="ICC_output.txt",append=TRUE)
    }


  }
  return(l_ICC)
}
```

```
l_ICC <- calculate_item_ICC_values(d_total, "low-ICCs for all participants")

l_ICC$Item = colnames(select(d_total,Q_E_HLA1:Q_E_UAI4)) # Add name code for each item
pander(l_ICC, caption = "All participants - ICC values for 100 items")
```

Table 1: All participants - ICC values for 100 items

i	icc	Item
1	0.7506	Q_E_HLA1
2	0.7577	Q_E_HLA2
3	0.7156	Q_E_HLA3
4	0.5686	Q_E_HLA4
5	0.6458	Q_E_HLB1
6	0.8155	Q_E_HLB2
7	0.7026	Q_E_HLB3
8	0.8488	Q_E_HLB4
9	0.7591	Q_E_HLB5
10	0.5675	Q_E_HLB5_2
11	0.4055	Q_E_NA1
12	0.5327	Q_E_NA2
13	0.7288	Q_E_NA3
14	0.4966	Q_E_NA4
15	0.5209	Q_E_NA5
16	0.4635	Q_E_NB1
17	0.7277	Q_E_NB2
18	0.8528	Q_E_NB3
19	0.3759	Q_E_AAS1
20	0.7897	Q_E_AAS2
21	0.6844	Q_E_AAS3
22	0.7726	Q_E_AU1
23	0.7344	Q_E_AU2
24	0.7755	Q_E_AU3
25	0.807	Q_E_PF1

i	icc	Item
26	0.7896	Q_E_PF2
27	0.6396	Q_E_PF3
28	0.6991	Q_E_AL1
29	0.8054	Q_E_AL1_2
30	0.9298	Q_E_AL2
31	0.8562	Q_E_R_AL3
32	0.6755	Q_E_AL4
33	0.8508	Q_E_AL5
34	0.5919	Q_E_AS1
35	0.6738	Q_E_AS1_2
36	0.7441	Q_E_AS1_3
37	0.4957	Q_E_AS2
38	0.5962	Q_E_AS2_2
39	0.6385	Q_E_AS2_3
40	0.6621	Q_E_AS3
41	0.2437	Q_E_APP1
42	0.5443	Q_E_R_APP2
43	0.8694	Q_E_APP3
44	0.6658	Q_E_UAA1
45	0.718	Q_E_UAA2
46	2.16e-09	Q_E_R_UAA3
47	0.7908	Q_E_R_AE1
48	0.8637	Q_E_AE2
49	0.7883	Q_E_AE3
50	0.4901	Q_E_R_AE4
51	0.5232	Q_E_UE1
52	0.3805	Q_E_UE2
53	0.3422	Q_E_UE2_2
54	0.02967	Q_E_UE3
55	0.8349	Q_E_UT1
56	0.8263	Q_E_UT2
57	0.6534	Q_E_UT2_2
58	0.5399	Q_E_UT2_3
59	0.7475	Q_E_UT3
60	0.6546	Q_E_UAL1
61	0.2909	Q_E_UAL1_2
62	0.389	Q_E_UAL2
63	0.2366	Q_E_UAL3

i	icc	Item
64	0.7441	Q_E_UAL4
65	0.4101	Q_E_UAL5
66	0.6702	Q_E_UAL6
67	0.5337	Q_E_AA1
68	0.7144	Q_E_AA2
69	0.7272	Q_E_AA3
70	0.6413	Q_E_R_AC1
71	0.8755	Q_E_R_AC2
72	0.7595	Q_E_R_AC3
73	0.7681	Q_E_R_AC4
74	0.3314	Q_E_AI1
75	0.6715	Q_E_AI2
76	0.4293	Q_E_R_AI3
77	0.5317	Q_E_AI4
78	0.7204	Q_E_AT1
79	0.4292	Q_E_AT2
80	0.5406	Q_E_R_AT3
81	0.7709	Q_E_SP1
82	0.4598	Q_E_SP2
83	0.5197	Q_E_SP3
84	0.7907	Q_E_IIS1
85	0.6562	Q_E_IIS2
86	0.5545	Q_E_IIS3
87	0.5943	Q_E_IIS4
88	0.5683	Q_E_AEI1
89	0.6933	Q_E_AEI2
90	0.5594	Q_E_R_AEI3
91	0.4519	Q_E_AEI4
92	0.2404	Q_E_R_AEI5
93	0.6521	Q_E_UEP1
94	0.6285	Q_E_UEP2
95	0.725	Q_E_UEP3
96	0.6282	Q_E_UEP4
97	0.4018	Q_E_UAI1
98	0.4346	Q_E_UAI2
99	0.3613	Q_E_UAI3
100	0.05423	Q_E_UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC$icc),digits=4),round(sd(l_ICC$icc),digits=4),
           round(min(l_ICC$icc),digits=4),round(max(l_ICC$icc),digits=4))
# Calculate the grand mean, standard deviation,
# minimum and maximum values of ICC values of 100 items
description <- cbind(Variable, Value) # Descriptive statistics of ICC values of 100 items

# Print results
pander(description, caption = paste("All participants - Descriptive",
                                    "statistics of ICC values of 100 items"))
```

Table 2: All participants - Descriptive statistics of ICC values of 100 items

Variable	Value
Grand_mean	0.6111
SD	0.1917
Minimum	0
Maximum	0.9298

We also calculate item-level ICC values for only the recommended data rows.

```r
l_ICC_recommended_1 <- calculate_item_ICC_values(
  d1_recommended, "low-ICCs for recommended participants, first half", n=50)
  nrow(l_ICC_recommended_1)
```

```
## [1] 50
```

```r
l_ICC_recommended_2 <- calculate_item_ICC_values(
  d2_recommended, "low-ICCs for recommended participants, second half", n=50)

l_ICC_recommended <- rbind(l_ICC_recommended_1, l_ICC_recommended_2)

l_ICC_recommended$Item = colnames(select(d_total,Q_E_HLA1:Q_E_UAI4)) # Add name code for each item
pander(l_ICC_recommended, caption = "Recommended participants - ICC values for 100 items")
```

Table 3: Recommended participants - ICC values for 100 items

i	icc	Item
1	0.7944	Q_E_HLA1
2	0.7501	Q_E_HLA2
3	0.7163	Q_E_HLA3
4	0.5342	Q_E_HLA4
5	0.6591	Q_E_HLB1
6	0.8074	Q_E_HLB2
7	0.6976	Q_E_HLB3
8	0.8714	Q_E_HLB4
9	0.7592	Q_E_HLB5
10	0.5592	Q_E_HLB5_2
11	0.4085	Q_E_NA1
12	0.5423	Q_E_NA2
13	0.7023	Q_E_NA3
14	0.5193	Q_E_NA4
15	0.5153	Q_E_NA5
16	0.4515	Q_E_NB1
17	0.7168	Q_E_NB2
18	0.857	Q_E_NB3
19	0.3564	Q_E_AAS1
20	0.7881	Q_E_AAS2
21	0.682	Q_E_AAS3
22	0.7655	Q_E_AU1
23	0.7281	Q_E_AU2
24	0.7699	Q_E_AU3
25	0.8676	Q_E_PF1
26	0.7844	Q_E_PF2
27	0.6239	Q_E_PF3
28	0.6876	Q_E_AL1
29	0.8015	Q_E_AL1_2
30	0.9312	Q_E_AL2
31	0.8504	Q_E_R_AL3
32	0.6716	Q_E_AL4
33	0.9146	Q_E_AL5
34	0.578	Q_E_AS1
35	0.6883	Q_E_AS1_2
36	0.7339	Q_E_AS1_3

i	icc	Item
37	0.4687	Q_E_AS2
38	0.5682	Q_E_AS2_2
39	0.6156	Q_E_AS2_3
40	0.6568	Q_E_AS3
41	0.2343	Q_E_APP1
42	0.519	Q_E_R_APP2
43	0.8723	Q_E_APP3
44	0.7426	Q_E_UAA1
45	0.7088	Q_E_UAA2
46	2.031e-09	Q_E_R_UAA3
47	0.7794	Q_E_R_AE1
48	0.8691	Q_E_AE2
49	0.7956	Q_E_AE3
50	0.4693	Q_E_R_AE4
1	0.5597	Q_E_UE1
2	0.3804	Q_E_UE2
3	0.3523	Q_E_UE2_2
4	0.03509	Q_E_UE3
5	0.8328	Q_E_UT1
6	0.8239	Q_E_UT2
7	0.6731	Q_E_UT2_2
8	0.5362	Q_E_UT2_3
9	0.7494	Q_E_UT3
10	0.7112	Q_E_UAL1
11	0.3195	Q_E_UAL1_2
12	0.4947	Q_E_UAL2
13	0.1756	Q_E_UAL3
14	0.715	Q_E_UAL4
15	0.4151	Q_E_UAL5
16	0.6548	Q_E_UAL6
17	0.4931	Q_E_AA1
18	0.6942	Q_E_AA2
19	0.7539	Q_E_AA3
20	0.6218	Q_E_R_AC1
21	0.87	Q_E_R_AC2
22	0.7481	Q_E_R_AC3
23	0.7526	Q_E_R_AC4
24	0.3095	Q_E_AI1

i	icc	Item
25	0.6922	Q_E_AI2
26	0.5399	Q_E_R_AI3
27	0.5181	Q_E_AI4
28	0.7174	Q_E_AT1
29	0.4175	Q_E_AT2
30	0.5224	Q_E_R_AT3
31	0.7664	Q_E_SP1
32	0.4547	Q_E_SP2
33	0.5814	Q_E_SP3
34	0.8029	Q_E_IIS1
35	0.6594	Q_E_IIS2
36	0.5457	Q_E_IIS3
37	0.6309	Q_E_IIS4
38	0.5625	Q_E_AEI1
39	0.6845	Q_E_AEI2
40	0.5631	Q_E_R_AEI3
41	0.3936	Q_E_AEI4
42	0.2128	Q_E_R_AEI5
43	0.6614	Q_E_UEP1
44	0.5973	Q_E_UEP2
45	0.7329	Q_E_UEP3
46	0.6085	Q_E_UEP4
47	0.423	Q_E_UAI1
48	0.4374	Q_E_UAI2
49	0.3291	Q_E_UAI3
50	0.06915	Q_E_UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC_recommended$icc),digits=4),
           round(sd(l_ICC_recommended$icc),digits=4),
           round(min(l_ICC_recommended$icc),digits=4),
           round(max(l_ICC_recommended$icc),digits=4))
# Calculate the grand mean, standard deviation,
# minimum and maximum values of ICC values of 100 items
description <- cbind(Variable, Value) # Descriptive statistics of ICC values of 100 items
```

```r
# Print results
pander(description, caption = paste("Recommended participants - Descriptive",
                                    "statistics of ICC values of 100 items"))
```

Table 4: Recommended participants - Descriptive statistics of ICC values of 100 items For the assessment of the correlation between the English and German ASA Questionnaire, we followed Cicchetti's classification of ICC categories (Cicchetti 1994). Then we get the categories of ICC classifications and number of ICC values in classification category. **Please note: There is no simple way of displaying only the best (highest-ICC) German items for English items with multiple translations. Below, you will find the classification of all translations.**

Variable	Value
Grand_mean	0.6118
SD	0.1949
Minimum	0
Maximum	0.9312

```r
Classification <- c("Excellent","Good","Fair","Poor")
ICC_Range <- c("0.75-1.00","0.60-0.74","0.40-0.59","0-0.39")
# Categories of ICC classifications by Cicchetti (1994)
n_item <- length(l_ICC$icc) # Number of ICC values
round_ICC <- round(l_ICC$icc, digits=4) # Round ICC values
Number <- c(length(l_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(l_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(l_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(l_ICC[which(round_ICC>=0.00&round_ICC<=0.39),]$icc))
# Calculate number of ICC values in classification category
Percentage <- c(round(Number[1]/n_item,digits=4)*100, round(Number[2]/n_item,digits=4)*100,
                round(Number[3]/n_item,digits=4)*100, round(Number[4]/n_item,digits=4)*100)
# Calculate percentage of ICC values in classification category
ICC_category <- cbind(Classification,ICC_Range,Number,Percentage)

# Print results
pander(ICC_category, caption = "Categories of ICC classifications and
       number of ICC values in classification category for 100 items")
```

Table 5: Categories of ICC classifications and number of ICC values in classification category for 100 items

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	26	26
Good	0.60-0.74	29	29
Fair	0.40-0.59	26	26
Poor	0-0.39	13	13

References

Cicchetti, Domenic V. 1994. "Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology." *Psychological Assessment* 6 (4): 284. https://doi.org/10.1037/1040-3590.6.4.284.

Finch, W Holmes, Jocelyn E Bolin, and Ken Kelley. 2019. *Multilevel Modeling Using r.* Crc Press.