Dutch ASA Questionnaire Translation - Part 1: Translation and Formative Assessment - Round 3

Nele Albers

19 March, 2024

Introduction

This document presents statistical analyses of correlation between Dutch and English ASA questionnaires items.

For this third translation round, 20 questionnaire items needed to be re-translated. Several questions had multiple alternative Dutch translations, meaning that participants answered a total of 20 English questions, and 84 Dutch translations.

The code is based on the one by Kriss Tesink and Johan Hensman for the first two Dutch translation rounds: https://doi.org/10.4121/9db602be-f748-46d7-b75d-874a1a0c244f.v1.

Required files: Data/Final_ASA_Dutch_Round_3_anonym.csv.

Setup

Let's load the packages we need.

```
library(car)  # Package linear regression
library(dplyr) # Use select function
library(knitr) # Get markdown file
library(nlme) # Run multilevel linear models
library(pander) # For pandering tables
panderOptions("table.alignment.default","left")
```

And we set constants to use throughout.

```
NUM_ENGLISH_ITEMS = 20  # Number of English questionnaire items
```

Load Data

The raw data gathered from the online platform Qualtrics was first anonymized by using the "anonymize_and_preprocess_third_dutch_translation_round.py" Python code, which produced the output file "Data/Final_ASA_Dutch_Round_3_anonym.csv."

30 Bilingual participants with Dutch as their first and primary language and with English as a language they were fluent in rated a Human-ASA interaction by answering a block of 20 English ASA questionnaire items and a block of 84 Dutch translations (half of the participants first rated the English items and the other half first rated the Dutch items). Included in each block were 7 attention check questions, for a total of 14 attention checks. We removed data from participants who failed attention checks (in this case, all participants passed the attention checks), and then removed all irrelevant data, keeping only the answers to the 20 English questions and 84 Dutch questions.

Additionally, in the survey, there was a question asking the participant whether or not they recommend the use of their data for research. All 30 participants answered with yes to this question.

```
dataset <- read.csv("Data/Final_ASA_Dutch_Round_3_anonym.csv")

# Select scores of 20 English items and corresponding 84 Dutch translations
d1 <- as.data.frame(select(dataset, HLA2_1:UAI4_4,
                           English_HLA2:English_UAI4))
```

Analysis and Results

ICC values were computed for all 84 translations, and a random intercept model was applied to the dataset. This model consists of a fixed intercept (~1) and incorporates participant as a random intercept, denoted by random = ~1|id. In this context, 'id' refers to the unique participant code assigned to the 30 bilingual participants whose scores were used to determine the ICC values.

We calculated ICC as: $\rho_I = \frac{\tau^2}{\tau^2+\sigma^2}$ where σ^2 is the variance within the score of individual, and τ^2 is the variance between participants (Finch, Bolin, and Kelley 2019).The *getICC* function is used to calculate the ICC value.

```
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)
}
```

The *getLME* function was defined to execute a multilevel model and then derive the corresponding ICC value for said model using the *getICC* function. This function requires the input of scores for both languages and the participant's ID number.

```
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_Dutch<- data.frame(id, s_1, language= 1)
  # Transform Dutch 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_Dutch, Score_English)
  # Combine Dutch 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))
}
```

After defining the *getLME* function, ICC values were calculated for each of the 84 translated questionnaire

items.

```
# Initialize output of ICC values of 20 items
l_ICC <- data.frame(ItemID = double(), Item = character(), icc = double())

# Offset for the 84 alternative translations
Dutch_column_offset <- 84

# A dynamic offset to keep track of how many alternative questions have been seen
Alt_questions_offset <- 0

d0 <- data.frame(d1)
names <- colnames(d0)

# Go step by step to 20 re translated items of the ASA questionnaire, whereby
# i is the ASA questionnaire item number
for (i in 1:NUM_ENGLISH_ITEMS)
{
    score_English <- na.omit(data.frame(score=d0[,i + Dutch_column_offset]))
    # Calculate English score
    question_length <- nchar(names[i + Alt_questions_offset])
    question_name <- substring(names[i + Alt_questions_offset], 1,
                               question_length-2)
    # get the name of the current question
    while(grepl(question_name, names[i + Alt_questions_offset], fixed=TRUE))
    {
      # iterate over all sub-questions
      score_Dutch <- na.omit(data.frame(score=d0[,i + Alt_questions_offset]))

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

      #increase the Alt_question_offset for each alternative translation we see
      Alt_questions_offset <- Alt_questions_offset + 1

    }
    Alt_questions_offset <- Alt_questions_offset - 1
}


l_ICC$Item = colnames(select(d0,HLA2_1:UAI4_4))

pander(l_ICC,
       caption = "ICC values for 84 items and their various retranslations")
```

Table 1: ICC values for 84 items and their various retranslations

i	icc	Item
1	0.6717	HLA2_1
1	0.6899	HLA2_2
1	0.581	HLA2_3
2	0.5107	HLB2_1

i	icc	Item
2	0.4746	HLB2_2
2	0.3182	HLB2_3
2	0.2188	HLB2_4
3	0.4979	HLB4_1
3	0.5315	HLB4_2
3	0.424	HLB4_3
4	0.5678	NA2_1
4	0.5022	NA2_2
4	0.414	NA2_3
4	0.7071	NA2_4
4	0.5731	NA2_5
4	0.6822	NA2_6
5	0.143	NA5_1
5	0.3256	NA5_2
5	0.598	NA5_3
5	0.6973	NA5_4
5	0.5759	NA5_5
5	0.7946	NA5_6
6	0.2325	NB2_1
6	0.5238	NB2_2
6	0.6265	NB2_3
6	0.6276	NB2_4
7	0.07965	NB3_1
7	0.5515	NB3_2
7	0.1183	NB3_3
7	0.4954	NB3_4
8	0.5197	AAS2_1
8	0.779	AAS2_2
8	0.695	AAS2_3
8	0.6765	AAS2_4
8	0.5354	AAS2_5
9	0.6322	AS1_1
9	0.4365	AS1_2
9	0.3531	AS1_3
9	0.569	AS1_4
9	0.6716	AS1_5
10	0.321	AS3_1
10	0.6243	AS3_2
10	0.2955	AS3_3
11	0.3617	UAA1_1
11	0.4897	UAA1_2
11	0.4706	UAA1_3
11	0.4519	UAA1_4
12	0.3219	UAA3_1
12	0.5683	UAA3_2
12	0.5417	UAA3_3
12	0.3669	UAA3_4
13	0.3196	UAL2_1
13	0.5047	UAL2_2
13	0.3992	UAL2_3
13	0.6213	UAL2_4
13	0.5369	UAL2_5

i	icc	Item
14	0.3613	UAL5_1
14	0.5222	UAL5_2
14	0.4703	UAL5_3
14	0.4468	UAL5_4
15	0.4547	AA2_1
15	0.6568	AA2_2
15	0.293	AA2_3
15	0.2195	AA2_4
16	0.7658	AA3_1
16	0.7461	AA3_2
16	0.7182	AA3_3
16	0.7433	AA3_4
17	0.8126	IIS4_1
17	0.5147	IIS4_2
17	0.6666	IIS4_3
17	0.6136	IIS4_4
17	0.5668	IIS4_5
18	0.5385	AEI1_1
18	0.5075	AEI1_2
18	0.5172	AEI1_3
18	0.1212	AEI1_4
19	0.429	UEP1_1
19	0.3393	UEP1_2
19	0.539	UEP1_3
20	0.5426	UAI4_1
20	0.3659	UAI4_2
20	0.576	UAI4_3
20	0.5129	UAI4_4

For the assessment of the correlation between the English and Dutch questionnaire items, we followed Cicchetti's categorization of ICC values (Cicchetti 1994), classifying questionnaire items into poor, fair, good, and excellent.

```
#Create categorizations of ICC values, ranging from poor to excellent
poor <- data.frame(ItemID = double(), Item = character(), icc = double())
fair <- data.frame(ItemID = double(), Item = character(), icc = double())
good <- data.frame(ItemID = double(), Item = character(), icc = double())
excellent <- data.frame(ItemID = double(), Item = character(), icc = double())

for(i in 1:84){
  if(l_ICC$icc[i]>=0.75) {
    #If the ICC value is greater than 0.75, it is excellent
     excellent <- rbind(excellent, data.frame (i ,l_ICC$Item[i] ,
                                               icc = l_ICC$icc[i]))
  } else if(l_ICC$icc[i]>=0.60) {
    #If the ICC value is between 0.60 and 0.75, it is good
     good <- rbind(good, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  } else if(l_ICC$icc[i]>=0.4) {
    #If the ICC value is between 0.4 and 0.6, it is fair
     fair <- rbind(fair, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  } else {
    #If the ICC value is below 0.4, it is poor
```

```
    poor <- rbind(poor, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  }
}

pander(poor, caption = "ICC values for poor items")
```

Table 2: ICC values for poor items

i	l_ICC.Item.i.	icc
6	HLB2_3	0.3182
7	HLB2_4	0.2188
17	NA5_1	0.143
18	NA5_2	0.3256
23	NB2_1	0.2325
27	NB3_1	0.07965
29	NB3_3	0.1183
38	AS1_3	0.3531
41	AS3_1	0.321
43	AS3_3	0.2955
44	UAA1_1	0.3617
48	UAA3_1	0.3219
51	UAA3_4	0.3669
52	UAL2_1	0.3196
54	UAL2_3	0.3992
57	UAL5_1	0.3613
63	AA2_3	0.293
64	AA2_4	0.2195
77	AEI1_4	0.1212
79	UEP1_2	0.3393
82	UAI4_2	0.3659

```
pander(fair, caption = "ICC values for fair items")
```

Table 3: ICC values for fair items

i	l_ICC.Item.i.	icc
3	HLA2_3	0.581
4	HLB2_1	0.5107
5	HLB2_2	0.4746
8	HLB4_1	0.4979
9	HLB4_2	0.5315
10	HLB4_3	0.424
11	NA2_1	0.5678
12	NA2_2	0.5022
13	NA2_3	0.414
15	NA2_5	0.5731
19	NA5_3	0.598
21	NA5_5	0.5759
24	NB2_2	0.5238
28	NB3_2	0.5515
30	NB3_4	0.4954

i	l_ICC.Item.i.	icc
31	AAS2_1	0.5197
35	AAS2_5	0.5354
37	AS1_2	0.4365
39	AS1_4	0.569
45	UAA1_2	0.4897
46	UAA1_3	0.4706
47	UAA1_4	0.4519
49	UAA3_2	0.5683
50	UAA3_3	0.5417
53	UAL2_2	0.5047
56	UAL2_5	0.5369
58	UAL5_2	0.5222
59	UAL5_3	0.4703
60	UAL5_4	0.4468
61	AA2_1	0.4547
70	IIS4_2	0.5147
73	IIS4_5	0.5668
74	AEI1_1	0.5385
75	AEI1_2	0.5075
76	AEI1_3	0.5172
78	UEP1_1	0.429
80	UEP1_3	0.539
81	UAI4_1	0.5426
83	UAI4_3	0.576
84	UAI4_4	0.5129

```
pander(good, caption = "ICC values for good items")
```

Table 4: ICC values for good items

i	l_ICC.Item.i.	icc
1	HLA2_1	0.6717
2	HLA2_2	0.6899
14	NA2_4	0.7071
16	NA2_6	0.6822
20	NA5_4	0.6973
25	NB2_3	0.6265
26	NB2_4	0.6276
33	AAS2_3	0.695
34	AAS2_4	0.6765
36	AS1_1	0.6322
40	AS1_5	0.6716
42	AS3_2	0.6243
55	UAL2_4	0.6213
62	AA2_2	0.6568
66	AA3_2	0.7461
67	AA3_3	0.7182
68	AA3_4	0.7433
71	IIS4_3	0.6666
72	IIS4_4	0.6136

```
pander(excellent, caption = "ICC values for excellent items")
```

Table 5: ICC values for excellent items

i	l_ICC.Item.i.	icc
22	NA5_6	0.7946
32	AAS2_2	0.779
65	AA3_1	0.7658
69	IIS4_1	0.8126

Here, we analyse the distribution of the categorizations, and display the percentage of ICC values in each category.

```
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=2) # 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 all alternative Dutch translations for the 20 Englis
```

Table 6: Categories of ICC classifications and number of ICC values in classification category for all alternative Dutch translations for the 20 English items.

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	5	5.95
Good	0.60-0.74	19	22.62
Fair	0.40-0.59	40	47.62
Poor	0-0.39	20	23.81

Finally, we iterate over all 84 translations, and select the best possible translations in the cases where there were multiple alternative translations, resulting in a list of `NUM_ENGLISH_ITEMS` translated items which correspond to the `NUM_ENGLISH_ITEMS` English questionnaire items.

```
# Create a new list for the final 20 questions, where the highest scoring ICC
# value is selected in case of multiple translation alternatives.
best_translations <- data.frame(ItemID = double(), Item = character(),
                                icc = double())
```

```r
# Go step by step to 20 re translated items of the ASA questionnaire,
# whereby i is the ASA questionnaire item number
Alt_questions_offset <- 0
for (i in 1:NUM_ENGLISH_ITEMS)
{
  #get the name of the current question
  question_length <- nchar(l_ICC$Item[i+Alt_questions_offset])
  question_name <- substring(l_ICC$Item[i+Alt_questions_offset], 1,
                             question_length-2)

  best_ICC <- 0
  best_question_number <- 0L
  while(grepl(question_name, l_ICC$Item[i+Alt_questions_offset], fixed=TRUE))
  {
    if(best_ICC <= l_ICC$icc[i+Alt_questions_offset])
    {
      # if the current sub question is better than the current best,
      # we override the best current score with the current sub question
      best_ICC <- l_ICC$icc[i+Alt_questions_offset]
      best_question_number <- i + Alt_questions_offset
    }

    # Increase the Alt_question_offset for each alternative translation we see
    Alt_questions_offset <- Alt_questions_offset + 1

  }
  # After traversing all sub questions, add the best sub question to
  # best_translations and reset best_ICC value to 0
  Alt_questions_offset <- Alt_questions_offset - 1
  best_translations <- rbind(best_translations, data.frame (i,
  Item = l_ICC$Item[best_question_number],
  icc = l_ICC$icc[best_question_number]))
  best_ICC <- 0

}

pander(best_translations, caption = "ICC values for 20 retranslated questions")
```

Table 7: ICC values for 20 retranslated questions

i	Item	icc
1	HLA2_2	0.6899
2	HLB2_1	0.5107
3	HLB4_2	0.5315
4	NA2_4	0.7071
5	NA5_6	0.7946
6	NB2_4	0.6276
7	NB3_2	0.5515
8	AAS2_2	0.779
9	AS1_5	0.6716
10	AS3_2	0.6243
11	UAA1_2	0.4897

i	Item	icc
12	UAA3_2	0.5683
13	UAL2_4	0.6213
14	UAL5_2	0.5222
15	AA2_2	0.6568
16	AA3_1	0.7658
17	IIS4_1	0.8126
18	AEI1_1	0.5385
19	UEP1_3	0.539
20	UAI4_3	0.576

The process of categorization and distribution analysis is repeated for the `NUM_ENGLISH_ITEMS` best Dutch translations.

```
poor_bestValues <- data.frame(ItemID = double(), Item = character(),
                              icc = double())
fair_bestValues <- data.frame(ItemID = double(), Item = character(),
                              icc = double())
good_bestValues <- data.frame(ItemID = double(), Item = character(),
                              icc = double())
excellent_bestValues <- data.frame(ItemID = double(), Item = character(),
                                   icc = double())

for(i in 1:NUM_ENGLISH_ITEMS){
  if(best_translations$icc[i]>= 0.75) {
    #If the ICC value is greater than 0.75, it is excellent
     excellent_bestValues <- rbind(excellent_bestValues,
                                   data.frame (i ,best_translations$Item[i] ,
                                               icc = best_translations$icc[i]))
  } else if(best_translations$icc[i]>=0.60) {
    #If the ICC value is between 0.60 and 0.75, it is good
     good_bestValues <- rbind(good_bestValues,
                              data.frame (i, best_translations$Item[i],
                                          icc = best_translations$icc[i]))
  } else if(best_translations$icc[i]>=0.4) {
    #If the ICC value is between 0.4 and 0.6, it is fair
     fair_bestValues <- rbind(fair_bestValues,
                              data.frame (i, best_translations$Item[i],
                                          icc = best_translations$icc[i]))
  } else {
    #If the ICC value is below 0.4, it is poor
     poor_bestValues <- rbind(poor_bestValues,
                              data.frame (i, best_translations$Item[i],
                                          icc = best_translations$icc[i]))
  }
}

pander(poor_bestValues, caption = "ICC values for poor items")
```

Table 8: ICC values for poor items

ItemID	Item	icc

```
pander(fair_bestValues, caption = "ICC values for fair items")
```

Table 9: ICC values for fair items

i	best_translations.Item.i.	icc
2	HLB2_1	0.5107
3	HLB4_2	0.5315
7	NB3_2	0.5515
11	UAA1_2	0.4897
12	UAA3_2	0.5683
14	UAL5_2	0.5222
18	AEI1_1	0.5385
19	UEP1_3	0.539
20	UAI4_3	0.576

```
pander(good_bestValues, caption = "ICC values for good items")
```

Table 10: ICC values for good items

i	best_translations.Item.i.	icc
1	HLA2_2	0.6899
4	NA2_4	0.7071
6	NB2_4	0.6276
9	AS1_5	0.6716
10	AS3_2	0.6243
13	UAL2_4	0.6213
15	AA2_2	0.6568

```
pander(excellent_bestValues, caption = "ICC values for excellent items")
```

Table 11: ICC values for excellent items

i	best_translations.Item.i.	icc
5	NA5_6	0.7946
8	AAS2_2	0.779
16	AA3_1	0.7658
17	IIS4_1	0.8126

```
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(best_translations$icc) # Number of ICC values
round_ICC <- round(best_translations$icc, digits=2) # Round ICC values
Number <- c(length(best_translations[which
                                     (round_ICC>=0.75&round_ICC<=1),]$icc),
            length(best_translations[which
                                     (round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(best_translations[which
                                     (round_ICC>=0.40&round_ICC<=0.59),]$icc),
```

```
            length(best_translations[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 the newly (best)
       20 translated items")
```

Table 12: Categories of ICC classifications and number of ICC values in classification category for the newly (best) 20 translated items

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	4	20
Good	0.60-0.74	7	35
Fair	0.40-0.59	9	45
Poor	0-0.39	0	0

```
cat("Mean of 20 ICC values:", mean(best_translations$icc), "\n
Standard deviation of 20 ICC values:", sd(best_translations$icc))
```

```
## Mean of 20 ICC values: 0.6289161
##
## Standard deviation of 20 ICC values: 0.1020704
```

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.

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