

Round_2_ICC

Kriss Tesink and Johan Hensman

2023-05-29

Introduction

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

For this second translation round, 37 questionnaire items needed to be re translated. Several questions had multiple alternative Dutch translations, meaning that participants answered a total of 37 English questions, and 64 Dutch translations.

We use the following packages:

```
library(foreign) # Open various data files
library(nlme)    # Run multilevel linear models
library(car)     # Package linear regression
library(haven)   # Use read_sav fuction
library(dplyr)   # Use select function
library(knitr)   # Get markdown file
library(tinytex) # Use TeX environment
library(rtticles) # Use CTeX documents template
library(pander)  # For pandering tables
panderOptions("table.alignment.default", "left")
```

Data file Final_ASA_Dutch_Round_2_anonym.sav

The raw data gathered from the online platform Qualtrics was first anonymized by using the “anonymize_and_preprocess_second_dutch_translation_round.py” python code, which produced the output file “Final_ASA_Dutch_Round_2_anonym.sav”.

30 Bilingual participants with a Dutch Mother tongue and fluency in English rated a Human-ASA interaction by first answering a block of 37 English ASA questionnaire items, followed by a block of 64 Dutch translations. Included in each block were 7 attention check control 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 37 English questions and 64 Dutch questions. Additionally, in the survey, there was a question asking the participant whether or not they recommend the use of their data, so data was removed for participants who answered no to this question (although in this case, all 30 participants answered yes to this question).

```
dataset3 <- read_sav("Final_ASA_Dutch_Round_2_anonym.sav")
dataset3 <- dataset3[!(dataset3$Use_data == 1),]
#Importing data and removing rows where participants did not recommend the use
```

*#of their data."dataset3" contains ASA questions whose ICC values were "poor" or
 # "fair" in the previous round and their corresponding new translations. Several
 # questions have multiple alternative translations.*

```
d3 <- na.omit(data.frame(select(dataset3, AC_Dutch_1:English_UAI3)))
#Select only relevant question items
dataset <- as.data.frame(lapply(d3[1:115], function(y)
  as.numeric(gsub('[a-zA-Z]', '', y))))
#Transform data representation into numeric
```

```
i <- grep("AC",colnames(dataset))
# Find the column number of attention control questions
Atten <- c(-3,3,3,-3,0,3,-3,-3,3,3,-3,0,3,-3)
# Correct answers for the 14 attention control questions
x <- NULL # Row number of participant who failed the attention check
for (j in (1:nrow(dataset))){
  # Find participants who failed attention check in 'dataset'
  count <- 0
  # The number of incorrectly answered attention questions of each participant
  for (k in 1:14){
    if (as.numeric(dataset[[i[k]]][j])!=Atten[k])
      # Check whether each participant's
      # attention question answers are consistent with the correct answers
      count <- count+1
  }
  if (count>0)
    # Row number of the participant who failed more than zero
    # 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)
  dataset <- dataset[-x,] #Participants who failed attention check were excluded
```

```
d1 <- as.data.frame(select(dataset, Dutch_HLA:Dutch_UAI3,
  English_HLA1:English_UAI3))
# Select scores of 44 English items and corresponding Dutch translations
```

Analysis and results

ICC values were computed for all 64 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 $\text{random} \sim 1|\text{id}$. 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 [Finch2019multilevel]. 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 64 translated questionnaire items

```

l_ICC <- data.frame(ItemID = double(), Item = character(), icc = double())
# Initialize output of ICC values of 37 items

Dutch_column_offset <- 64
#Offset for the 64 alternative translations
Alt_questions_offset <- 0
#A dynamic offset to keep track of how many alternative questions have been seen
d0 <- data.frame(d1)
names <- colnames(d0)

for (i in 1:37)
  # Go step by step to 37 re translated items of the ASA questionnaire, whereby
  #i is the ASA questionnaire item number
{
  if(grepl("Alt", names[i + Alt_questions_offset], fixed=TRUE))

```

```

# Check if sub-questions exist for current question
{
  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]))

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

    Alt_questions_offset <- Alt_questions_offset + 1
    #increase the Alt_question_offset for each alternative translation we see
  }
  Alt_questions_offset <- Alt_questions_offset - 1
} else
{
  #if there are no alternative translations, we calculate the ICC value
  score_Dutch <- na.omit(data.frame(score=d0[,i + Alt_questions_offset]))
  # Select scores of Dutch version of ASAQ item i

  score_English <- na.omit(data.frame(score=d0[,i + Dutch_column_offset]))
  # Select scores of English version of ASAQ items i
  iccScore <- getLME(score_Dutch, score_English)
  l_ICC <- rbind(l_ICC, data.frame (i, icc = iccScore))

  # Calculated ICC and add it to the list of ICC values,
  # with ID number of the ASA questionnaire item
}
}

l_ICC$Item = colnames(select(d0,Dutch_HLA:Dutch_UAI3))

pander(l_ICC,
       caption = "ICC values for 37 items and their various retranslations")

```

Table 1: ICC values for 37 items and their various retranslations
 For the assessment of the correlation between the English and Dutch questionnaire items, we followed Cicchetti’s categorization of ICC values [Cicchetti1994guidelines], classifying questionnaire items into poor, fair, good, and excellent.

i	icc	Item
1	0.4252	Dutch_HLA

i	icc	Item
2	0.1967	Dutch_HLB1_Alt1
2	0.3898	Dutch_HLB1_Alt2
3	0.5491	Dutch_HLB2
4	0.3608	Dutch_NA1
5	0.6298	Dutch_NA2_Alt1
5	0.4569	Dutch_NA2_Alt2
6	0.6054	Dutch_NA3
7	0.296	Dutch_NA4_Alt1
7	0.4726	Dutch_NA4_Alt2
8	0.603	Dutch_NB1
9	0.37	Dutch_NB2_Alt1
9	0.4159	Dutch_NB2_Alt2
10	0.3371	Dutch_NB3_Alt1
10	0.4326	Dutch_NB3_Alt2
11	0.5327	Dutch_AAS1
12	0.6055	Dutch_AAS2_Alt1
12	0.5745	Dutch_AAS2_Alt2
12	0.7264	Dutch_AAS2_Alt3
12	0.7904	Dutch_AAS2_Alt4
13	0.6966	Dutch_AU1_Alt1
13	0.6695	Dutch_AU1_Alt2
14	0.6732	Dutch_AL1_Alt1
14	0.5069	Dutch_AL1_Alt2
15	0.5784	Dutch_AS1_Alt1
15	0.4415	Dutch_AS1_Alt2
15	0.4606	Dutch_AS1_Alt3
16	0.3757	Dutch_AS2_Alt1
16	0.2989	Dutch_AS2_Alt2
16	0.595	Dutch_AS2_Alt3
17	0.4843	Dutch_UAA1_Alt1
17	0.5218	Dutch_UAA1_Alt2
18	0.05263	Dutch_UAA2
19	0.6942	Dutch_AE1_Alt1
19	4.912e-09	Dutch_AE1_Alt2
20	0.7005	Dutch_AE2
21	0.3781	Dutch_UE1_Alt1
21	0.8083	Dutch_UE1_Alt2
22	0.6662	Dutch_UT1_Alt1
22	0.4821	Dutch_UT1_Alt2
23	0.559	Dutch_UAL1_Alt1
23	0.4725	Dutch_UAL1_Alt2
24	0.5466	Dutch_UAL2_Alt1
24	0.4805	Dutch_UAL2_Alt2
25	0.355	Dutch_AA1_Alt1
25	0.368	Dutch_AA1_Alt2
26	0.5846	Dutch_AA2
27	0.6247	Dutch_AT1_Alt1
27	0.5626	Dutch_AT1_Alt2
28	0.632	Dutch_SP
29	0.7753	Dutch_IIS1_Alt1
29	0.808	Dutch_IIS1_Alt2
30	0.6422	Dutch_IIS2_Alt1

i	icc	Item
30	0.7088	Dutch_IIS2_Alt2
31	0.5775	Dutch_IIS3_Alt1
31	0.6341	Dutch_IIS3_Alt2
32	0.5529	Dutch_IIS4
33	0.5169	Dutch_AEI1_Alt1
33	0.3999	Dutch_AEI1_Alt2
34	0.5204	Dutch_UEP1_Alt1
34	0.198	Dutch_UEP1_Alt2
35	0.6482	Dutch_UAI1
36	0.7664	Dutch_UAI2
37	0.4189	Dutch_UAI3

```

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())
#Create categorizations of ICC values, ranging from poor to excellent

for(i in 1:64){
  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
2	Dutch_HLB1_Alt1	0.1967
3	Dutch_HLB1_Alt2	0.3898
5	Dutch_NA1	0.3608
9	Dutch_NA4_Alt1	0.296
12	Dutch_NB2_Alt1	0.37
14	Dutch_NB3_Alt1	0.3371
28	Dutch_AS2_Alt1	0.3757
29	Dutch_AS2_Alt2	0.2989
33	Dutch_UAA2	0.05263
35	Dutch_AEI1_Alt2	4.912e-09

i	l_ICC.Item.i.	icc
37	Dutch_UE1_Alt1	0.3781
45	Dutch_AA1_Alt1	0.355
46	Dutch_AA1_Alt2	0.368
59	Dutch_AEI1_Alt2	0.3999
61	Dutch_UEP1_Alt2	0.198

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

Table 3: ICC values for fair items

i	l_ICC.Item.i.	icc
1	Dutch_HLA	0.4252
4	Dutch_HLB2	0.5491
7	Dutch_NA2_Alt2	0.4569
10	Dutch_NA4_Alt2	0.4726
13	Dutch_NB2_Alt2	0.4159
15	Dutch_NB3_Alt2	0.4326
16	Dutch_AAS1	0.5327
18	Dutch_AAS2_Alt2	0.5745
24	Dutch_AL1_Alt2	0.5069
25	Dutch_AS1_Alt1	0.5784
26	Dutch_AS1_Alt2	0.4415
27	Dutch_AS1_Alt3	0.4606
30	Dutch_AS2_Alt3	0.595
31	Dutch_UAA1_Alt1	0.4843
32	Dutch_UAA1_Alt2	0.5218
40	Dutch_UT1_Alt2	0.4821
41	Dutch_UAL1_Alt1	0.559
42	Dutch_UAL1_Alt2	0.4725
43	Dutch_UAL2_Alt1	0.5466
44	Dutch_UAL2_Alt2	0.4805
47	Dutch_AA2	0.5846
49	Dutch_AT1_Alt2	0.5626
55	Dutch_IIS3_Alt1	0.5775
57	Dutch_IIS4	0.5529
58	Dutch_AEI1_Alt1	0.5169
60	Dutch_UEP1_Alt1	0.5204
64	Dutch_UAI3	0.4189

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

Table 4: ICC values for good items

i	l_ICC.Item.i.	icc
6	Dutch_NA2_Alt1	0.6298
8	Dutch_NA3	0.6054
11	Dutch_NB1	0.603
17	Dutch_AAS2_Alt1	0.6055

i	l_ICC.Item.i.	icc
19	Dutch_AAS2_Alt3	0.7264
21	Dutch_AU1_Alt1	0.6966
22	Dutch_AU1_Alt2	0.6695
23	Dutch_AL1_Alt1	0.6732
34	Dutch_AE1_Alt1	0.6942
36	Dutch_AE2	0.7005
39	Dutch_UT1_Alt1	0.6662
48	Dutch_AT1_Alt1	0.6247
50	Dutch_SP	0.632
53	Dutch_IIS2_Alt1	0.6422
54	Dutch_IIS2_Alt2	0.7088
56	Dutch_IIS3_Alt2	0.6341
62	Dutch_UAI1	0.6482

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

Table 5: ICC values for excellent items Here, we analyse the distribution of the categorizations, and display the percentage ICC values in each category.

i	l_ICC.Item.i.	icc
20	Dutch_AAS2_Alt4	0.7904
38	Dutch_UE1_Alt2	0.8083
51	Dutch_IIS1_Alt1	0.7753
52	Dutch_IIS1_Alt2	0.808
63	Dutch_UAI2	0.7664

```
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 the newly 37 translations
and all the alternative translations")
```


Table 6: Categories of ICC classifications and number of ICC values in classification category for the newly 37 translations and all the alternative translations. Finally, we iterate over all 64 translations, and select the best possible translations in the cases where there were multiple alternative translations, resulting in a list of 37 translated items which correspond to the 37 English questionnaire items.

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	5	7.81
Good	0.60-0.74	17	26.56
Fair	0.40-0.59	28	43.75
Poor	0-0.39	14	21.88

```
best_translations <- data.frame(ItemID = double(), Item = character(),
                                icc = double())
#Create a new list for the final 37 questions, where the highest scoring ICC
#value is selected in case of multiple translation alternatives.
Alt_questions_offset <- 0
for (i in 1:37)
  # Go step by step to 37 re translated items of the ASA questionnaire,
  #whereby i is the ASA questionnaire item number
  {
    if(grepl("Alt", l_ICC$Item[i+Alt_questions_offset], fixed=TRUE))
      # Check if sub-questions exist for current question
      {
        question_length <- nchar(l_ICC$Item[i+Alt_questions_offset])
        question_name <- substring(l_ICC$Item[i+Alt_questions_offset], 1,
                                   question_length-2)
        #get the name of the current question

        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$icc[i+Alt_questions_offset]
            best_question_number <- i + Alt_questions_offset
          }
          Alt_questions_offset <- Alt_questions_offset + 1
          #Increase the Alt_question_offset for each alternative translation we see
        }
        #After traversing all sub questions, add the best sub question to
        #best_translations and reset best_ICC value to 0
        Alt_questions_offset <- Alt_questions_offset - 1
        best_translations <- rbind(best_translations, data.frame (i,
                                                                    Item = l_ICC$Item[best_question_number],
                                                                    icc = l_ICC$icc[best_question_number]))
        best_ICC <- 0
      } else
```

```

{
  #if there are no Alternative sub-questions, simply add the current question
  #to best_translations
  best_translations <- rbind(best_translations, data.frame (i,
    Item = l_ICC$Item[i+Alt_questions_offset],
    icc = l_ICC$icc[i+Alt_questions_offset]))
}
}
pander(best_translations, caption = "ICC values for 37 retranslated questions")

```

Table 7: ICC values for 37 retranslated questions

i	Item	icc
1	Dutch_HLA	0.4252
2	Dutch_HLB1_Alt2	0.3898
3	Dutch_HLB2	0.5491
4	Dutch_NA1	0.3608
5	Dutch_NA2_Alt1	0.6298
6	Dutch_NA3	0.6054
7	Dutch_NA4_Alt2	0.4726
8	Dutch_NB1	0.603
9	Dutch_NB2_Alt2	0.4159
10	Dutch_NB3_Alt2	0.4326
11	Dutch_AAS1	0.5327
12	Dutch_AAS2_Alt4	0.7904
13	Dutch_AU1_Alt1	0.6966
14	Dutch_AL1_Alt1	0.6732
15	Dutch_AS1_Alt1	0.5784
16	Dutch_AS2_Alt3	0.595
17	Dutch_UAA1_Alt2	0.5218
18	Dutch_UAA2	0.05263
19	Dutch_AE1_Alt1	0.6942
20	Dutch_AE2	0.7005
21	Dutch_UE1_Alt2	0.8083
22	Dutch_UT1_Alt1	0.6662
23	Dutch_UAL1_Alt1	0.559
24	Dutch_UAL2_Alt1	0.5466
25	Dutch_AA1_Alt2	0.368
26	Dutch_AA2	0.5846
27	Dutch_AT1_Alt1	0.6247
28	Dutch_SP	0.632
29	Dutch_IIS1_Alt2	0.808
30	Dutch_IIS2_Alt2	0.7088
31	Dutch_IIS3_Alt2	0.6341
32	Dutch_IIS4	0.5529
33	Dutch_AEI1_Alt1	0.5169
34	Dutch_UEP1_Alt1	0.5204
35	Dutch_UAI1	0.6482
36	Dutch_UAI2	0.7664
37	Dutch_UAI3	0.4189

```
#Create categorizations of ICC values, ranging from poor to excellent
```

The process of categorization and distribution analysis is repeated for the 37 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 1:37){
  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

i	best_translations.Item.i.	icc
2	Dutch_HLB1_Alt2	0.3898
4	Dutch_NA1	0.3608
18	Dutch_UAA2	0.05263
25	Dutch_AA1_Alt2	0.368

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

Table 9: ICC values for fair items

i	best_translations.Item.i.	icc
1	Dutch_HLA	0.4252
3	Dutch_HLB2	0.5491
7	Dutch_NA4_Alt2	0.4726
9	Dutch_NB2_Alt2	0.4159
10	Dutch_NB3_Alt2	0.4326
11	Dutch_AAS1	0.5327
15	Dutch_AS1_Alt1	0.5784
16	Dutch_AS2_Alt3	0.595
17	Dutch_UAA1_Alt2	0.5218
23	Dutch_UAL1_Alt1	0.559
24	Dutch_UAL2_Alt1	0.5466
26	Dutch_AA2	0.5846
32	Dutch_IIS4	0.5529
33	Dutch_AEI1_Alt1	0.5169
34	Dutch_UEP1_Alt1	0.5204
37	Dutch_UAI3	0.4189

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

Table 10: ICC values for good items

i	best_translations.Item.i.	icc
5	Dutch_NA2_Alt1	0.6298
6	Dutch_NA3	0.6054
8	Dutch_NB1	0.603
13	Dutch_AU1_Alt1	0.6966
14	Dutch_AL1_Alt1	0.6732
19	Dutch_AE1_Alt1	0.6942
20	Dutch_AE2	0.7005
22	Dutch_UT1_Alt1	0.6662
27	Dutch_AT1_Alt1	0.6247
28	Dutch_SP	0.632
30	Dutch_IIS2_Alt2	0.7088
31	Dutch_IIS3_Alt2	0.6341
35	Dutch_UAI1	0.6482

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

Table 11: ICC values for excellent items

i	best_translations.Item.i.	icc
12	Dutch_AAS2_Alt4	0.7904
21	Dutch_UE1_Alt2	0.8083
29	Dutch_IIS1_Alt2	0.808
36	Dutch_UAI2	0.7664

```

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)
37 translated items")

```

Table 12: Categories of ICC classifications and number of ICC values in classification category for the newly (best) 37 translated items

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	4	10.81
Good	0.60-0.74	13	35.14
Fair	0.40-0.59	16	43.24
Poor	0-0.39	4	10.81

```

cat("Mean of 37 ICC values:", mean(best_translations$icc), "\n
Standard deviation of 37 ICC values:", sd(best_translations$icc))

```

```

## Mean of 37 ICC values: 0.5698316
##
## Standard deviation of 37 ICC values: 0.1493677

```