Qualtrics Rating Preproc & Prelim Viz & Stats

Yuchao Wang

6/6/2020

Preprocessing

- Using a `.csv` file, `forWordSeparation_cleaned`, to create a *word* by *participant* matrix with behavioral ratings (*emotional arousal* or *literariness*) from Qualtrics survey outputs.
- Specifically, `forWordSeparation_cleaned.csv` is read in to create `template`, which is used to search for the rating per word per participant.

```
#Clean up csv files containing all rated words in survey----
template <- read.csv(file = "forWordSeparation_cleaned.csv",
                     header = F, sep = ",", stringsAsFactors = FALSE)
template <- t(template)
colnames(template) <- template[1, ]
template <- template[-1, ]
template <- as.data.frame(template) #data type converted to matrix after transpose, due
template <- gather(template, "qnLabel", "word", 1:17)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
template <- template[!(template$word == ""), ]
template <- template[ , c("word","qnLabel")]
template$qnLabel <- trimws(template$qnLabel)
template$qnLabel <- as.factor(template$qnLabel)
template$qnLabel <- factor(template$qnLabel, levels = c("3.1","5.1","7.1","9.1",
                                                        "11.1","13.1","15.1","17.1","19.
                                                        "28.1","30.1","32.1","34.1","36.
qnLabel <- levels(template$qnLabel)


#Match emo results----
emoResult <- read.csv(file="Verhalen_Emotional+Arousal_October+1%2C+2019_13.27.csv",
                  header = TRUE, sep = ",", stringsAsFactors = FALSE)
emoResult <- emoResult[-c(1,2), ] #get rid of title row
emoResult <- emoResult[-c(1:9), ] #get rid of preview Test results
emoResult <- emoResult[-c(15:17,28), ] #get rid of incomplete results


totalParticipant <- nrow(emoResult)
totalWord <- nrow(template)
emoResultClean <- template


for (indexResponse in 1:totalParticipant) {
  for (indexWord in 1:totalWord) {
    searchWord <- template[indexWord,1]
    searchQn <- template[indexWord,2]
    resultCol <- indexResponse+2
    emoResultClean[indexWord,resultCol] <- NA #initialize a new column
      for (possibleRating in 1:7) {
        if (!is.na(emoResultClean[indexWord,resultCol])) {
          break
        }
        currentColName <- paste ("Q", searchQn, "_", possibleRating, sep = "", collapse
        if (grepl(searchWord, emoResult[indexResponse,currentColName], fixed = TRUE)) {
            emoResultClean[indexWord,resultCol] <- substr(currentColName,
                                                          nchar(currentColName), nchar(c
        }
        else {
          if ((possibleRating == 7) & (is.na(emoResultClean[indexWord,resultCol]))) {
            emoResultClean[indexWord,resultCol] <- NA
          }
        }
      }
  }
}


#Match lit results----
litResult <- read.csv(file="Verhalen_Literariness_October+1%2C+2019_13.28.csv",
                  header = TRUE, sep = ",", stringsAsFactors = FALSE)
litResult <- litResult[-c(1,2), ] #get rid of title row
litResult <- litResult[-c(1:10), ] #get rid of preview Test results


totalParticipant <- nrow(litResult)
totalWord <- nrow(template)
litResultClean <- template
```

```
for (indexResponse in 1:totalParticipant) {
  for (indexWord in 1:totalWord) {
    searchWord <- template[indexWord,1]
    searchQn <- template[indexWord,2]
    resultCol <- indexResponse+2
    litResultClean[indexWord,resultCol] <- NA #initialize a new column
      for (possibleRating in 1:2) {
        if (!is.na(litResultClean[indexWord,resultCol])) {
          break
        }
        currentColName <- paste ("Q", searchQn, "_", possibleRating, sep = "", collapse
        if (grepl(searchWord, litResult[indexResponse,currentColName], fixed = TRUE)) {
            litResultClean[indexWord,resultCol] <- substr(currentColName,
                                                          nchar(currentColName), nchar(c
        }
        else {
          if ((possibleRating == 2) & (is.na(litResultClean[indexWord,resultCol]))) {
            litResultClean[indexWord,resultCol] <- NA
          }
        }
      }
  }
}

#Save clean files----
saveRDS(emoResultClean, file = "emoResultClean.rds")
saveRDS(litResultClean, file = "litResultClean.rds")
saveRDS(template, file = "template.rds")
```

Preliminary Visualization (not included)

- see `MatchPraatTiming.rmd` for final viz
- Visualize the Qualtrics ratings per word across participants for each of the two stories (`DM` and `DH`).

Stats - Intraclass Correlation Coefficient

- To check how well participants agree with each other on ratings.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
## 
##     expand, pack, unpack
```

```
emoResultCleanICC <- matrix(data = NA, nrow = 0, ncol = 27)
for (index in 1:2354) {
  emoResultCleanICC <- rbind(emoResultCleanICC, emoResultClean[index, 3:29])
}
for (j in 1:27) {
    emoResultCleanICC[,j] <- as.numeric(emoResultCleanICC[,j])
}
ICC(emoResultCleanICC, missing = FALSE, alpha = 0.05)
```

```
## Call: ICC(x = emoResultCleanICC, missing = FALSE, alpha = 0.05)
## 
## Intraclass correlation coefficients 
##                          type  ICC  F  df1   df2 p lower bound upper bound
## Single_raters_absolute   ICC1 0.18  7 2353 61204 0        0.17        0.19
## Single_random_raters     ICC2 0.19 12 2353 61178 0        0.16        0.23
## Single_fixed_raters      ICC3 0.28 12 2353 61178 0        0.27        0.29
## Average_raters_absolute ICC1k 0.86  7 2353 61204 0        0.85        0.86
## Average_random_raters   ICC2k 0.86 12 2353 61178 0        0.83        0.89
## Average_fixed_raters    ICC3k 0.91 12 2353 61178 0        0.91        0.92
## 
##  Number of subjects = 2354     Number of Judges =  27
```

```
litResultCleanICC <- matrix(data = NA, nrow = 0, ncol = 27)
for (index in 1:2354) {
  litResultCleanICC <- rbind(litResultCleanICC, litResultClean[index, 3:29])
}
for (j in 1:27) {
    litResultCleanICC[,j] <- as.numeric(litResultCleanICC[,j])
}
ICC(litResultCleanICC, missing = FALSE, alpha = 0.05)
```

```
## Call: ICC(x = litResultCleanICC, missing = FALSE, alpha = 0.05)
## 
## Intraclass correlation coefficients 
##                          type  ICC    F  df1   df2 p lower bound
## Single_raters_absolute   ICC1 0.24  9.3 2353 61204 0        0.22
## Single_random_raters     ICC2 0.24 10.6 2353 61178 0        0.22
## Single_fixed_raters      ICC3 0.26 10.6 2353 61178 0        0.25
## Average_raters_absolute ICC1k 0.89  9.3 2353 61204 0        0.89
## Average_random_raters   ICC2k 0.89 10.6 2353 61178 0        0.88
## Average_fixed_raters    ICC3k 0.91 10.6 2353 61178 0        0.90
##                         upper bound
## Single_raters_absolute         0.25
## Single_random_raters           0.25
## Single_fixed_raters            0.28
## Average_raters_absolute        0.90
## Average_random_raters          0.90
## Average_fixed_raters           0.91
##
##  Number of subjects = 2354     Number of Judges =  27
```

Stats - Corr between Emo and Lit

- To check if *emotional arousal* and *literariness* are distinct and can serve as regressors in separate fMRI GLMs.
- Specifically, checked both the correlation using 1. the original word-by-word mean values and 2. the rollmean with two preceding and two succeeding words

```
#Simple linear regression ANOVA of Emo on Lit ratings, with and without boxcar smoothi
library('zoo')
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
emoDHTrend <- cbind(emoDHTrend, rollmean(emoDHTrend$mean, k = 5, fill = NA, align = "cen
names(emoDHTrend)[14] <- "rollmean"
litDHTrend <- cbind(litDHTrend, rollmean(litDHTrend$mean, k = 5, fill = NA, align = "cen
names(litDHTrend)[14] <- "rollmean"
#emo_lit_DHmodel <- lm(emoDHTrend$mean ~ litDHTrend$mean)
#emo_lit_DHboxedmodel <- lm(emoDHTrend$rollmean ~ litDHTrend$rollmean)
#anova(emo_lit_DHmodel)
#anova(emo_lit_DHboxedmodel)
cor.test(emoDHTrend$mean, litDHTrend$mean, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  emoDHTrend$mean and litDHTrend$mean
## t = 11.077, df = 1234, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2491395 0.3506013
## sample estimates:
##      cor
## 0.300721
```

```
emoDMTrend <- cbind(emoDMTrend, rollmean(emoDMTrend$mean, k = 5, fill = NA, align = "cer
names(emoDMTrend)[14] <- "rollmean"
litDMTrend <- cbind(litDMTrend, rollmean(litDMTrend$mean, k = 5, fill = NA, align = "cer
names(litDMTrend)[14] <- "rollmean"
#emo_lit_DMmodel <- lm(emoDMTrend$mean ~ litDMTrend$mean)
#emo_lit_DMboxedmodel <- lm(emoDMTrend$rollmean ~ litDMTrend$rollmean)
#anova(emo_lit_DMmodel)
#anova(emo_lit_DMboxedmodel)
cor.test(emoDMTrend$mean, litDMTrend$mean, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  emoDMTrend$mean and litDMTrend$mean
## t = -0.78249, df = 1115, p-value = 0.4341
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08196979  0.03527649
## sample estimates:
##         cor
## -0.02342721
```