Analysis\_IRR

1. Set up: libraries, directory and dataset
2. Identify excerpts across codings that share the same text w/in ind participants, using three methods
3. Exact match
4. Excerpt start and end characters within some margin (1000 chars)
5. Excerpt start >= dat2 start - marg and excerpt end <= dat2 end + marg

table(dat1$participant %in% dat2$participant)

##   
## TRUE   
## 570

table(dat2$participant %in% dat1$participant)

##   
## TRUE   
## 743

unique(dat1$participant[!(dat1$participant %in% dat2$participant)])

## character(0)

unique(dat2$participant[!(dat2$participant %in% dat1$participant)])

## character(0)

# Set up vectors to store matches  
dat1$matched00 <- NA   
dat1$matched01 <- NA   
dat1$matched02 <- NA   
dat1$matched03 <- NA   
dat1$matched04 <- NA   
  
# Start with exact matches   
for (i in 1:dim(dat1)[1]){  
 x <- dat1$exc\_raw\_append[i]  
 out <- which(dat2$exc\_raw\_append==x)  
 matches <- concatenate(dat2$uniq\_crime[out])  
 dat1$matched00[i] <- matches  
}  
  
  
## Check whether w/in N chars of both start and end points  
marg <- 1000  
for (i in 1:dim(dat1)[1]){  
 x1 <- dat2$exc\_start\_min[i]  
 x2 <- dat2$exc\_end\_max[i]  
 part <- dat2$participant[i]  
 out <- which(dat1$exc\_start\_min >= x1 - marg & dat1$exc\_start\_min <= x1 + marg &   
 dat1$exc\_end\_max >= x2 - marg & dat1$exc\_end\_max <= x2 + marg &  
 dat1$participant==part)  
 matches <- concatenate(dat2$uniq\_crime[out])  
 dat1$matched01[i] <- matches  
}  
  
  
## Check whether w/in N chars of both start and end points  
marg <- 2000  
for (i in 1:dim(dat1)[1]){  
 x1 <- dat2$exc\_start\_min[i]  
 x2 <- dat2$exc\_end\_max[i]  
 part <- dat2$participant[i]  
 out <- which(dat1$exc\_start\_min >= x1 - marg & dat1$exc\_start\_min <= x1 + marg &   
 dat1$exc\_end\_max >= x2 - marg & dat1$exc\_end\_max <= x2 + marg &  
 dat1$participant==part)  
 matches <- concatenate(dat2$uniq\_crime[out])  
 dat1$matched02[i] <- matches  
}  
  
  
## Check whether contained w/in min - N marg and max + N marg  
marg <- 1000  
for (i in 1:dim(dat1)[1]){  
 x1 <- dat2$exc\_start\_min[i]  
 x2 <- dat2$exc\_end\_max[i]  
 part <- dat2$participant[i]  
 out <- which(dat1$exc\_start\_min >= x1 - marg &   
 dat1$exc\_end\_max <= x2 + marg &  
 dat1$participant==part)  
 matches <- concatenate(dat2$uniq\_crime[out])  
 dat1$matched03[i] <- matches  
}  
  
  
## Check whether contained w/in min - N marg and max + N marg of longest  
marg <- 500  
for (i in 1:dim(dat1)[1]){  
 x1 <- dat2$exc\_longest\_start[i]  
 x2 <- dat2$exc\_longest\_end[i]  
 part <- dat2$participant[i]  
 out <- which(dat1$exc\_longest\_start >= x1 - marg &   
 dat1$exc\_longest\_end <= x2 + marg &  
 dat1$participant==part)  
 matches <- concatenate(dat2$uniq\_crime[out])  
 dat1$matched04[i] <- matches  
}

1. Assess quality of each type of match

## Proportion of dat1 excerpts matched  
prop.table(table(dat1$matched00==""))

##   
## FALSE TRUE   
## 0.154386 0.845614

prop.table(table(dat1$matched01==""))

##   
## FALSE TRUE   
## 0.3368421 0.6631579

prop.table(table(dat1$matched02==""))

##   
## FALSE TRUE   
## 0.4403509 0.5596491

prop.table(table(dat1$matched03==""))

##   
## FALSE TRUE   
## 0.5368421 0.4631579

prop.table(table(dat1$matched04==""))

##   
## FALSE TRUE   
## 0.4087719 0.5912281

1. Create dataset of matches that includes both codings. Leave in duplicate matches. dat00 = exact matches dat01 = matches w/in 1000 chars of both start and end location dat02 = matches w/in 2000 chars of both start and end location dat03 = matches contained w/in start - 1000 and end + 1000

## Split list of matches  
dat00 <- dat1 %>%  
 separate(col = matched00, into = 'match1', sep = " ", extra = 'merge', fill = 'right')  
dat01 <- dat1 %>%  
 separate(col = matched01, into = c('match1', 'match2'), sep = " ", extra = 'merge', fill = 'right')  
dat02 <- dat1 %>%  
 separate(col = matched02, into = c('match1', 'match2','match3', 'match4','match5'), sep = " ", extra = 'merge', fill = 'right')  
dat03 <- dat1 %>%  
 separate(col = matched03, into = paste0('match',seq(1,18)), sep = " ", extra = 'merge', fill = 'right')  
dat04 <- dat1 %>%  
 separate(col = matched04, into = paste0('match',seq(1,4)), sep = " ", extra = 'merge', fill = 'right')  
  
## Reshape  
dat00 <- dat00 %>%   
 select(uniq\_crime, participant, match1, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun) %>%  
 rename(uniq\_crime\_dat2 = match1) %>%  
 subset(uniq\_crime\_dat2!="")  
  
dat01 <- dat01 %>%   
 select(uniq\_crime, participant, match1:match2, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun) %>%   
 gather(key = 'n\_match', value = "uniq\_crime\_dat2", match1:match2) %>%   
 subset(uniq\_crime\_dat2!="")  
  
dat02 <- dat02 %>%   
 select(uniq\_crime, participant, match1:match5, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun) %>%   
 gather(key = 'n\_match', value = "uniq\_crime\_dat2", match1:match5) %>%  
 subset(uniq\_crime\_dat2!="")  
  
dat03 <- dat03 %>%   
 select(uniq\_crime, participant, match1:match18, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun) %>%   
 gather(key = 'n\_match', value = "uniq\_crime\_dat2", match1:match18) %>%  
 subset(uniq\_crime\_dat2!="")  
  
dat04 <- dat04 %>%   
 select(uniq\_crime, participant, match1:match4, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun) %>%   
 gather(key = 'n\_match', value = "uniq\_crime\_dat2", match1:match4) %>%  
 subset(uniq\_crime\_dat2!="")  
  
## Merge in dat2 codings  
  
dat2 <- dat2 %>%  
 select(uniq\_crime, participant, deonto\_uc, conseq\_uc, human\_uc, dehuman\_uc,  
 rel\_crime, phy\_cri, lethal\_pun, let\_phy\_pun)   
setnames(dat2, colnames(dat2), paste0(colnames(dat2), "\_dat2"))  
  
dat00 <- merge(dat00, dat2, by.x = c('uniq\_crime\_dat2'), by.y = c('uniq\_crime\_dat2'), all.x=T)  
dat01 <- merge(dat01, dat2, by.x = c('uniq\_crime\_dat2'), by.y = c('uniq\_crime\_dat2'), all.x=T)  
dat02 <- merge(dat02, dat2, by.x = c('uniq\_crime\_dat2'), by.y = c('uniq\_crime\_dat2'), all.x=T)  
dat03 <- merge(dat03, dat2, by.x = c('uniq\_crime\_dat2'), by.y = c('uniq\_crime\_dat2'), all.x=T)  
dat04 <- merge(dat04, dat2, by.x = c('uniq\_crime\_dat2'), by.y = c('uniq\_crime\_dat2'), all.x=T)  
  
## write exact matches to csv  
t1 <- subset(dat1, select = c('uniq\_crime', 'participant', 'exc\_raw\_append'))  
t <- merge(dat00, t1, by = c('uniq\_crime', 'participant'), all.x = T)  
write.csv(t, "../02\_Data/02\_Clean/irr\_exact.xlsx")  
  
## subset of deonto mismatches  
t2 <- subset(t, t$deonto\_uc!=t$deonto\_uc\_dat2)  
write.csv(t2, "../02\_Data/02\_Clean/irr\_exact\_deontodiff.csv", fileEncoding = "UTF-8")  
  
## subset of empathy by either coder  
t <- subset(dat1, dat1$human\_uc==1, select = c('uniq\_crime', 'participant', 'deonto\_uc', 'conseq\_uc',  
 'dehuman\_uc', 'human\_uc', 'exc\_raw\_append'))  
write.csv(t, "../02\_Data/02\_Clean/dat1\_human.csv", fileEncoding = "UTF-8")  
  
t <- readit('../02\_Data/02\_Clean/crime\_set2\_clean.xlsx')

## File guessed to be xls/xlsx (Excel) ("../02\_Data/02\_Clean/crime\_set2\_clean.xlsx")

t <- subset(t, t$human\_uc==1, select = c('uniq\_crime', 'participant', 'deonto\_uc', 'conseq\_uc',  
 'dehuman\_uc', 'human\_uc', 'exc\_raw\_append'))  
write.csv(t, "../02\_Data/02\_Clean/dat2\_human.csv", fileEncoding = "UTF-8")

1. Look at IRR

out <- data.frame('var' = c('rel\_crime', 'phy\_cri', 'lethal\_pun', 'let\_phy\_pun', 'deonto\_uc', 'conseq\_uc', 'human\_uc', 'dehuman\_uc'),  
 'exact' = NA,  
 'all\_mar\_500' = NA,  
 'all\_mar\_1000' = NA,  
 'all\_within\_500' = NA,  
 'longest\_mar\_500' = NA,  
 stringsAsFactors = F)  
  
vars <- as.character(out$var)  
  
for (i in 1:length(vars)){  
 var\_dat1 <- dat00[,vars[i]]  
 var\_dat2 <- dat00[,paste0(vars[i],'\_dat2')]  
 tab <- table(var\_dat1, var\_dat2)  
agree <- sum(diag(tab))  
diag(tab) <- NA  
disagree <- sum(tab, na.rm=T)  
out[i,'exact'] <- agree/(agree + disagree)  
}  
  
for (i in 1:length(vars)){  
 var\_dat1 <- dat01[,vars[i]]  
 var\_dat2 <- dat01[,paste0(vars[i],'\_dat2')]  
 tab <- table(var\_dat1, var\_dat2)  
agree <- sum(diag(tab))  
diag(tab) <- NA  
disagree <- sum(tab, na.rm=T)  
out[i,'all\_mar\_500'] <- agree/(agree + disagree)  
}  
  
for (i in 1:length(vars)){  
 var\_dat1 <- dat02[,vars[i]]  
 var\_dat2 <- dat02[,paste0(vars[i],'\_dat2')]  
 tab <- table(var\_dat1, var\_dat2)  
agree <- sum(diag(tab))  
diag(tab) <- NA  
disagree <- sum(tab, na.rm=T)  
out[i,'all\_mar\_1000'] <- agree/(agree + disagree)  
}  
  
for (i in 1:length(vars)){  
 var\_dat1 <- dat03[,vars[i]]  
 var\_dat2 <- dat03[,paste0(vars[i],'\_dat2')]  
 tab <- table(var\_dat1, var\_dat2)  
agree <- sum(diag(tab))  
diag(tab) <- NA  
disagree <- sum(tab, na.rm=T)  
out[i,'all\_within\_500'] <- agree/(agree + disagree)  
}  
  
  
for (i in 1:length(vars)){  
 var\_dat1 <- dat04[,vars[i]]  
 var\_dat2 <- dat04[,paste0(vars[i],'\_dat2')]  
 tab <- table(var\_dat1, var\_dat2)  
agree <- sum(diag(tab))  
diag(tab) <- NA  
disagree <- sum(tab, na.rm=T)  
out[i,'longest\_mar\_500'] <- agree/(agree + disagree)  
}  
  
prop\_matched <- c('prop\_matched',  
 length(unique(dat00$uniq\_crime))/dim(dat1)[1],  
 length(unique(dat01$uniq\_crime))/dim(dat1)[1],  
 length(unique(dat02$uniq\_crime))/dim(dat1)[1],  
 length(unique(dat03$uniq\_crime))/dim(dat1)[1],  
 length(unique(dat04$uniq\_crime))/dim(dat1)[1])  
out <- rbind(out, prop\_matched)  
vars <- out$var; out$var=NULL  
out <- apply(out, 2, as.numeric)  
rownames(out) <- vars  
xtable(out, digits = 2)

## % latex table generated in R 4.3.1 by xtable 1.8-4 package  
## % Mon Dec 11 12:54:02 2023  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{rrrrrr}  
## \hline  
## & exact & all\\_mar\\_500 & all\\_mar\\_1000 & all\\_within\\_500 & longest\\_mar\\_500 \\   
## \hline  
## rel\\_crime & 0.86 & 0.37 & 0.36 & 0.39 & 0.41 \\   
## phy\\_cri & 0.98 & 0.49 & 0.49 & 0.50 & 0.47 \\   
## lethal\\_pun & 0.95 & 0.85 & 0.83 & 0.84 & 0.86 \\   
## let\\_phy\\_pun & 0.86 & 0.62 & 0.63 & 0.66 & 0.67 \\   
## deonto\\_uc & 0.52 & 0.60 & 0.58 & 0.56 & 0.60 \\   
## conseq\\_uc & 0.88 & 0.65 & 0.64 & 0.63 & 0.62 \\   
## human\\_uc & 0.88 & 0.87 & 0.87 & 0.87 & 0.86 \\   
## dehuman\\_uc & 0.93 & 0.90 & 0.91 & 0.89 & 0.91 \\   
## prop\\_matched & 0.15 & 0.34 & 0.44 & 0.54 & 0.41 \\   
## \hline  
## \end{tabular}  
## \end{table}