# Add Praat Timing to Qualtrics Ratings

Yuchao Wang

6/6/2020

## Match word ratings with onset

- Match Qualtrics ratings per word (for 1st person perspective stories) with the onset time of that word in both 1st and 3rd person perspective story recordings.
- Praat timing files adjusted to eliminate mismatches due to formatting or transcription differences.
- Add a column `onset`.

## Add mean and SE

- Add rating mean and SE as additional columns to matrix, and clean mismatched rows
- Save resulted matrix per story (`DM` and `DH`) per perspective (`O` and `2V`) per rating (`emo` and `lit`) as `.csv` files.

```
stderr <- function(x) {
  sd(x, na.rm = TRUE)/sqrt(length(x[!is.na(x)]))
}

addMNSECol <- function(resultwTiming, colNum) {
  for (colIndex in 3: colNum) {
    resultwTiming[, colIndex] <- as.numeric(resultwTiming[, colIndex])
  }
  resultwTiming <- transform(resultwTiming, MN = rowMeans(resultwTiming[ ,3:colNum], na.
  rowNum <- nrow(resultwTiming)
  for (rowIndex in 1:rowNum) {
    resultwTiming[rowIndex, colNum+4] <- stderr(resultwTiming[rowIndex, 3:colNum])
  }
  names(resultwTiming)[colNum+4] <- 'SE'
  return(resultwTiming)
}

emoOGwTiming <- addMNSECol(emoOGwTiming, colNumEmo)
emo2VwTiming <- addMNSECol(emo2VwTiming, colNumEmo)
litOGwTiming <- addMNSECol(litOGwTiming, colNumLit)
```

```r
lit2VwTiming <- addMNSECol(lit2VwTiming, colNumLit)



#WRITE OUTPUT FILES
emoDHOFinal <- matrix(data = NA, nrow = 0, ncol = colNumEmo+4)
emoDMOFinal <- matrix(data = NA, nrow = 0, ncol = colNumEmo+4)
emoDH2VFinal <- matrix(data = NA, nrow = 0, ncol = colNumEmo+4)
emoDM2VFinal <- matrix(data = NA, nrow = 0, ncol = colNumEmo+4)
litDHOFinal <- matrix(data = NA, nrow = 0, ncol = colNumLit+4)
litDMOFinal <- matrix(data = NA, nrow = 0, ncol = colNumLit+4)
litDH2VFinal <- matrix(data = NA, nrow = 0, ncol = colNumLit+4)
litDM2VFinal <- matrix(data = NA, nrow = 0, ncol = colNumLit+4)

#This function cleans mismatched entries and write CSV output files.
cleanOutput <- function(totalWord, colNum, rawData, outputFile, outputName) {
  if (totalWord == 1237) {
    for (index in 1:totalWord) {
      if ((rawData[index, colNum+2] == "mismatch") | (rawData[index, colNum+2] == "0"))
        next
      }
      else {
        outputFile <- rbind(outputFile, rawData[index, ])
      }
    }
  }
  else {
    for (index in 1:totalWord) {
      if ((rawData[index+1237, colNum+2] == "mismatch") | (rawData[index+1237, colNum+2]
        next
      }
      else {
        outputFile <- rbind(outputFile, rawData[index+1237, ])
      }
    }
  }
  write.csv(outputFile, file = paste("thesis/", outputName, ".csv", sep=""))
  return(outputFile)
}


emoDHOFinal <- cleanOutput(1237, colNumEmo, emoOGwTiming, emoDHOFinal, "emoDHOFinal")
emoDMOFinal <- cleanOutput(1117, colNumEmo, emoOGwTiming, emoDMOFinal, "emoDMOFinal")
emoDH2VFinal <- cleanOutput(1237, colNumEmo, emo2VwTiming, emoDH2VFinal, "emoDH2VFinal")
```

```
emoDM2VFinal <- cleanOutput(1117, colNumEmo, emo2VwTiming, emoDM2VFinal, "emoDM2VFinal")
litDHOFinal <- cleanOutput(1237, colNumLit, litOGwTiming, litDHOFinal, "litDHOFinal")
litDMOFinal <- cleanOutput(1117, colNumLit, litOGwTiming, litDMOFinal, "litDMOFinal")
litDH2VFinal <- cleanOutput(1237, colNumLit, lit2VwTiming, litDH2VFinal, "litDH2VFinal")
litDM2VFinal <- cleanOutput(1117, colNumLit, lit2VwTiming, litDM2VFinal, "litDM2VFinal")
```

## Preliminary Visualization (not included)

- With mean word ratings against word onset time, with error bars.

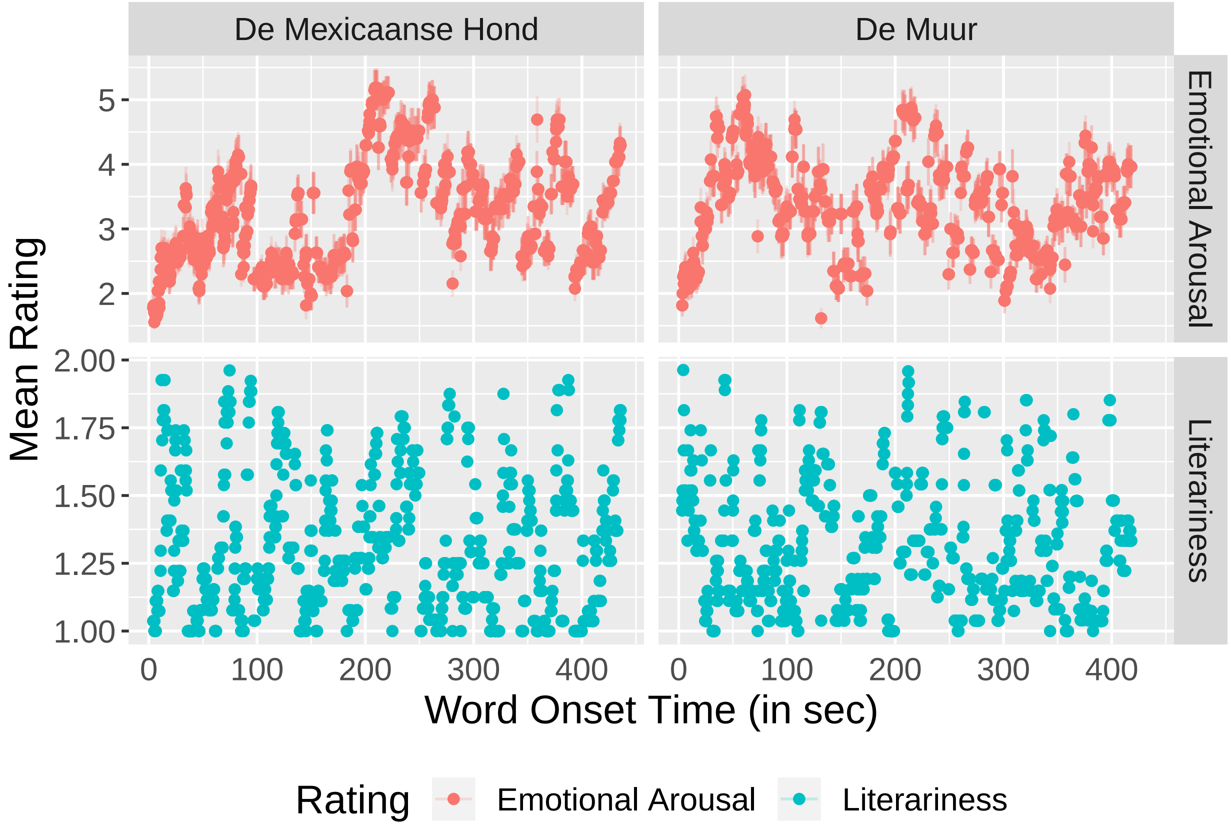
## Final Visualization

- To be used in manuscript
- `AllOriginal_eventMarked.csv` created by manually concatenating `emoDHOFinal_eventMarked`, `emoDMOFinal_eventMarked`, `litDHOFinal_eventMarked`, `litDMOFinal_eventMarked`.
- Overall plots of ratings against onset, per story, per type of rating (`emo` and `lit`).
- For 1. per word ratings 2. per semantic event ratings.
- SE for literariness is remapped to 0 because its a binary decision and SE does not make sense.

```
allRatingMarked <- read.csv(file = "thesis/AllOriginal_eventMarked.csv", header = T, sep

#correct for interpretation of literariness SE by changing all lit SE to 0
for (i in 1:nrow(allRatingMarked)) {
  if (allRatingMarked$Rating[i] == "Literariness") {
    allRatingMarked$SE[i] = 0
  }
}

limits <- aes(ymax = allRatingMarked$MN + allRatingMarked$SE,
              ymin = allRatingMarked$MN - allRatingMarked$SE)
word_plot <- ggplot(data = allRatingMarked, aes(x = Onset, y = MN, color = Rating)) +
    geom_point(size=1.5) +
    geom_errorbar(limits, width = 0.1, alpha = 0.2) +
    xlab("Word Onset Time (in sec)") +
    ylab("Mean Rating") +
    theme(text = element_text(size=15),  axis.ticks.x=element_blank(), legend.position=
    facet_grid(Rating ~ Story, scales = "free_y")
event_plot <- ggplot(data = allRatingMarked, aes(x = Onset_event, y = MN_event, color =
    geom_step(direction = "hv", size=0.5) +
    xlab("Event Onset Time (in sec)") +
    ylab("Mean Rating") +
    theme(text = element_text(size=15),  axis.ticks.x=element_blank(), legend.position=
    facet_grid(Rating ~ Story, scales = "free_y")
```

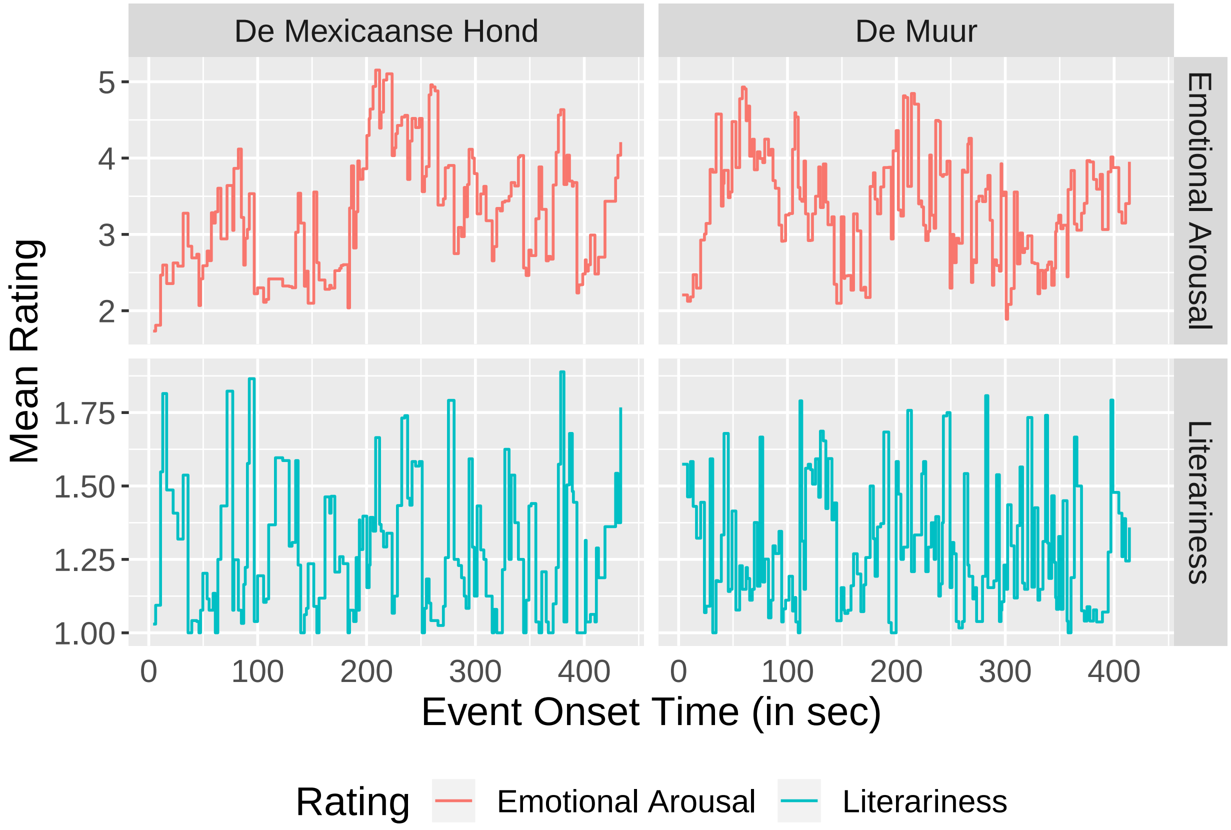
```
word_plot
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_errorbar).
```



```
event_plot
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```



```
ggsave("~/Desktop/allWordRating.png", plot = word_plot, device = png(),
    scale = 1, width = 400, height = 240, units = c("mm"),
    dpi = 300, limitsize = TRUE)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_errorbar).
```

```
ggsave("~/Desktop/allEventRating.png", plot = event_plot, device = png(),
    scale = 1, width = 400, height = 240, units = c("mm"),
    dpi = 300, limitsize = TRUE)
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```

### Stats - Emo ~ Lit (stories concatenated)

```
allEmoWordRating <- unlist(subset(allRatingMarked, Rating == "Emotional Arousal", select
allLitWordRating <- unlist(subset(allRatingMarked, Rating == "Literariness", select = "M
allEmoEventRating <- unlist(subset(allRatingMarked, Rating == "Emotional Arousal", selec
allLitEventRating <- unlist(subset(allRatingMarked, Rating == "Literariness", select = "
cor.test(allEmoWordRating, allLitWordRating, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  allEmoWordRating and allLitWordRating
## t = 8.1896, df = 2341, p-value = 4.258e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1272542 0.2059923
## sample estimates:
##       cor
## 0.1668893
```

```
cor.test(allEmoEventRating, allLitEventRating, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  allEmoEventRating and allLitEventRating
## t = 2.8498, df = 323, p-value = 0.004655
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04864803 0.26095741
## sample estimates:
##       cor
## 0.1566114
```