

```
## -----  
##  
## Script name: 05-study-2.R  
##  
## Project: Tracing Thick Concepts  
##  
##  
##  
##  
##  
##  
## -----  
##  
## Notes:  
## Not all of the analyses  
## and results in this  
## script were used in the  
## final paper  
##  
## -----  
  
## -----  
##### 1 Libraries #####  
## -----  
library(quanteda)  
library(dplyr)  
library(openxlsx)  
library(factoextra) # hcla, loads ggplot2  
library(stringr)  
library(reshape2)  
library(xtable)  
library(Ckmeans.1d.dp) # univariate k-means  
library(utc)  
library(nnet)  
library(MNLpred)  
library(scales)  
library(grid)  
library(gridExtra)  
library(viridis)  
library(caret)  
rm(list=ls())  
  
## -----  
## 2 Set working directory ###  
## -----  
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

```

getwd()

## -----
##### 3 UDFs #####
## -----
abbrv <- function(x, width = 200) lapply(strwrap(x, width, simplify = FALSE), paste,
collapse="\n")

## -----
##### 4 Load data #####
## -----
kdata <- read.xlsx('../input/study2.xlsx') %>% .[-ncol(.)] %>% as_tibble()

## -----
##### 5 Analysis #####
## -----
## Complete Type var
index <- c(grep('.', kdata$Type), length(kdata$Type)+1)
reps <- na.omit(abs(lag(index)-index))
repl <- rep(na.omit(kdata$Type), reps)
length(repl) == nrow(kdata)
cbind(repl, kdata$Type)
kdata$Type <- repl
write.table(kdata, file = '../input/study2_results.csv', sep = ',', row.names = F)

## -----
## NOTE:
kdata <- filter(kdata, !Adjective%in%c('awful', 'disgusting'))

## -----
## Hierarchical cluster analysis & dendogram
varsel <- grep('Eval\\. ', colnames(kdata), value = T)
clusdata <- kdata[, varsel] %>% as.data.frame()
rownames(clusdata) <- kdata$Adjective
?cluster::maxSE
res <- eclust(clusdata, FUNcluster = 'hclust', hc_method = 'ward.D', hc_metric = 'euclidean',
seed = 2, stand = F,
      gap_maxSE = list(d.power = 2, method = 'firstmax', spaceH0 = 'scaledPCA'), k = 3)
col_vec <- ifelse(kdata$Type %in% c('Value-Associated', 'Descriptive'), 'darkgrey', 'black')
col_vec <- col_vec[res$order]
p <- fviz_dend(res, phylo_layout = 'layout.gem', k_colors = "black",
  label_cols = col_vec, cex = 0.9, rect = T, lower_rect = -8,
  #horiz = T,
  lwd = 0.5,
  main = "
  )
p

```

```

p_coord <- ggplot_build(p)$data[[4]]
p <- p +
  #geom_text(aes(x = p_coord$xmax, y = p_coord$ymin, label = 1:2, hjust = 2, vjust = -.9)) +
  #geom_text(aes(x = p_coord$xmin[1], y = p_coord$ymin[1], label = 'Cluster:', hjust = -.2,
vjust = -.9)) +
  theme(plot.title = element_blank())
p
ggsave(p, filename = '../output/paper/paper_study2_dendogramm.pdf', height = 6, width =
9)

```

```

## -----
## plot for presentation
col_vec[col_vec == 'black'] <- alpha('red', 0.8)
col_vec[col_vec == 'darkgrey'] <- 'black'
p <- fviz_dend(res, phylo_layout = 'layout.gem', k_colors = "black",
  label_cols = col_vec, cex = 0.9, rect = T, lower_rect = -8,
  #horiz = T,
  lwd = 0.5,
  main = "
)
p2 <- p +
  #scale_y_continuous(limits = c(-8, 8)) +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    plot.background = element_rect(fill="transparent", color = NA),
    panel.background = element_rect(fill="transparent", color = NA)
  )
p
ggsave(p2, filename = '../output/paper/PRESI_paper_study2_dendogramm.pdf', height = 6,
width = 10, bg = 'transparent')

```

```

## -----
## Univariate k-means (does not work better)
## Might be removed in the final version, since it is not part of the results
kdata
uvres <- Ckmeans.1d.dp(kdata$Eval.Weight)

aggr <- cbind(kdata, uvres = uvres$cluster)

p <- ggplot(aggr, aes(x = Eval.Weight, y = Adjective)) +
  geom_text(aes(label = uvres)) +
  #scale_x_continuous(limits = c(-.5,.5)) +
  facet_grid(Type~., scales = 'free_y', space = "free_y", drop = T) +
  labs(

```

```

y = 'Target Ajective',
x = 'Eval.Weight',
caption = abbrev('NTC: Neg. Thick Concepts; NVAC: Neg. Value-Associated Concepts; NThin:
Neg. Thin Concepts; DC: Descriptive Concepts; PThin: Pos. Thin Concepts; PVAC: Pos. Value-
Associated Concepts; PTC: Pos. Thick Concepts.', width = 75)
)
p

## -----
## Multinomial probit model
kdata <- kdata %>% mutate(Type = ifelse(Type == "Thick and Thin Non-Moral", "Thick Non-
Moral", Type))
kdata$Type2 <- relevel(factor(kdata$Type), ref = "Value-Associated")
kdata$Type2 <- factor(kdata$Type2, levels = c("Value-Associated", "Descriptive", "Thick
Non-Moral", "Thick Moral", "Thin"))
range(kdata$Eval.Weight)
kdata$Type3 <- as.numeric(kdata$Type2)
#m1 <- multinom(as.numeric(Type2) ~ Eval.Weight, data = kdata, Hess = T)
m1 <- multinom(Type3 ~ Eval.Weight, data = kdata[c(3,11)], Hess = T)
tempdata <- data.frame(model.frame(m1))
## Predictions
preds <- mnl_pred_ova(
  model = m1,
  data = tempdata,
  x = 'Eval.Weight',
  by = 0.01,
  seed = 176,
  nsim = 100,
  probs = c(0.025, 0.975)
)
## Plot loop for plot panel
# recode data
preds$plotdata <- preds$plotdata %>%
  mutate(
    Type3 = as.character(Type3),
    Type2 = recode(Type3, "1" = "Value-Associated", "2" = "Descriptive", "3" = "Thick Non-
Moral", "4" = "Thick Moral", "5" = "Thin")
  )
unique(preds$plotdata$Type2)
plist <- list()
for(i in 1:length(levels(kdata$Type2))){
  index <- levels(kdata$Type2)[i]
  plist[[i]] <- ggplot(filter(preds$plotdata, Type2==index), aes(x=Eval.Weight, y=mean, ymin =
lower, ymax = upper)) +
    geom_ribbon(alpha = 0.1, colour = NA) +
    geom_line(linetype = c('solid', 'dashed', '12345678', 'dotdash', 'dotted')[i]) +
    facet_wrap(Type2~., ncol = 2) +

```

```

theme_bw() +
theme(
  plot.title = element_text(face = 'bold'),
  plot.background = element_rect(fill=NA, color = NA)
) +
scale_y_continuous(limits = c(0,1), labels = scales::percent_format(accuracy = 1)) +
labs(
  x="",
  y=""
)
}
plist[[1]]
plist[[2]]
plist[[3]]
plist[[4]]
plist[[5]]

plist[[6]] <-
  ggplot(preds$plotdata, aes(x=Eval.Weight, y=mean, group=factor(Type2, levels = c("Value-
Associated", "Descriptive", "Thick Non-Moral", "Thick Moral", "Thin")))) +
  #geom_ribbon(alpha = 0.1, colour = NA) +
  geom_line(aes(lty=Type2)) +
  theme_bw() +
  theme(
    plot.title = element_text(face = 'bold'),
    legend.position = 'none',
    plot.background = element_rect(fill=NA, color = NA)
  ) +
  scale_y_continuous(limits = c(0,1), labels = scales::percent_format(accuracy = 1)) +
  scale_linetype_manual(values = c('dashed', '12345678', 'dotdash', 'dotted', 'solid')) +
  labs(
    x="",
    y=""
    #lty='Class'
  )
}
plist[[6]]
p <- grid.arrange(
  textGrob(abbrev("Predicted Predicted Probabilities for Class Membership based on Eval",
width = 60), x=0, y=0.9, vjust = 1, hjust = 0, gp = gpar(fontface = "bold", cex = 1)),
  plist[[1]], plist[[2]], plist[[3]], plist[[4]], plist[[5]], plist[[6]],
  ncol = 2,
  bottom = textGrob("Eval", x = 0.55),
  left = textGrob('Probability', rot = 90, vjust = 1),
  layout_matrix = rbind(c(1,1),
                        c(2,3),
                        c(4,5),
                        c(6,7)),

```

```

heights = c(0.5,2,2,2)
)
p

ggsave(p, filename = '../output/paper/paper_study2_class_membership_MNL.pdf', width =
5, height = 7)

p <- grid.arrange(
  plist[[1]], plist[[2]], plist[[3]], plist[[4]], plist[[5]], plist[[6]],
  ncol = 2,
  bottom = textGrob("Eval", x = 0.55),
  left = textGrob('Probability', rot = 90, vjust = 1),
  layout_matrix = rbind(c(2,3)-1,
                        c(4,5)-1,
                        c(6,7)-1),
  heights = c(2,2,2)
)
p

ggsave(p, filename =
'../output/paper/paper_study2_class_membership_MNL_wo_title.pdf', width = 5, height =
6)

## -----
## For presentations
p <- grid.arrange(
  textGrob(abbrev("Predicted Predicted Probabilities for Class Membership based on Eval",
width = 100), x=0, y=0.9, vjust = 1, hjust = 0, gp = gpar(fontface = "bold", cex = 1)),
  plist[[1]], plist[[2]], plist[[3]], plist[[4]], plist[[5]], plist[[6]],
  ncol = 3,
  bottom = textGrob("Eval", x = 0.55),
  left = textGrob('Probability', rot = 90, vjust = 1),
  layout_matrix = rbind(c(1,1,NA),
                        c(2,3,4),
                        c(5,6,7)),
  heights = c(0.2,2,2)
)
p

ggsave(p, filename = '../output/paper/PRESI_paper_study2_class_membership_MNL.pdf',
width = 8, height = 5, bg = 'transparent')

p <- grid.arrange(
  plist[[1]], plist[[2]], plist[[3]], plist[[4]], plist[[5]], plist[[6]],
  ncol = 3,
  bottom = textGrob("Eval", x = 0.55),

```

```

left = textGrob('Probability', rot = 90, vjust = 1),
layout_matrix = rbind(c(2,3,4)-1,
                      c(5,6,7)-1),
heights = c(2,2)
)
p

ggsave(p, filename =
'../output/paper/PRESI_paper_study2_class_membership_MNL_wo_title.pdf', width = 8,
height = 4, bg = 'transparent')

## -----
## Evaluation metric for the model
response <- predict(m1, newdata = kdata, type = "probs")
colnames(response) <- c("Value-Associated", "Descriptive", "Thick Non-Moral", "Thick
Moral", "Thin")
max_prob <- colnames(response)[apply(response,1,which.max)]
max_prob <- factor(max_prob, levels = c("Value-Associated", "Descriptive", "Thick Non-
Moral", "Thick Moral", "Thin"))
levels(max_prob)
levels(kdata$Type2)
cm <- confusionMatrix(max_prob, reference = kdata$Type2)
rownames(cm$byClass) <- gsub('Class\\:', '', rownames(cm$byClass))
cm$byClass
rws <- seq(1, (nrow(cm$byClass)-1), by = 2)
col <- rep("\\rowcolor[gray]{0.95}", length(rws))
tab <- cm$byClass[,c(1:2,5:7,11)]
colnames(tab) <- gsub('Balanced ', '', colnames(tab))
comment <- c(paste("\\hline\\", "{\\footnotesize Overall Accuracy: 0.5349 (CI:
0.3765/0.6882); Kappa: 0.4065; No-information rate: 0.2356.}\\n", sep = ""))
print(xtable(tab, digits = 4,
             label = "tab:evalMetricsMultinom",
             caption = "Model evaluation metrics by concept class for multinomial logistic
regression."), comment = F,
      add.to.row = list(pos = as.list(c(rws, nrow(tab))), command = c(col, comment)),
      size='\\footnotesize', caption.placement = "top",
      hline.after = c(-1,0))
cm$overall

## -----
## First version of the panel, might be removed in the final version
p <- ggplot(preds$plotdata, aes(x=Eval.Weight, y=mean, ymin = lower, ymax = upper)) +
  geom_ribbon(alpha = 0.1, colour = NA) +
  geom_line() +
  facet_wrap(Type2~., ncol = 2) +
  theme_bw() +

```

```

theme(
  plot.title = element_text(face = 'bold')
) +
scale_y_continuous(limits = c(0,1), labels = percent_format(accuracy = 1)) +
labs(
  title = abbrev('Predicted Probabilities for Class Membership based on Eval', width = 50),
  x = 'Eval',
  y = 'Probability'
)
p

## -----
## Eval vs non-eval logit model
kdata$evalType <- factor(ifelse(kdata$Type %in% c('Value-Associated', 'Descriptive'), 'non-
eval', 'eval'))
kdata$evalType <- relevel(kdata$evalType, ref = 'non-eval')
kdata <- filter(kdata, !Adjective == c('awful', 'disgusting'))
range(kdata$Eval.Weight)
m2 <- glm(evalType ~ Eval.Weight, data = kdata, family = "binomial")
## Predictions
newData <- data.frame(Eval.Weight = seq(0, max(kdata$Eval.Weight), 0.01))
newData <- cbind(newData, predict(m2, newData, type = 'link', se = T))
newData <- within(newData, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})
## Plot
p <- ggplot(newData, aes(x = Eval.Weight, y = PredictedProb)) +
  geom_ribbon(aes(ymin = LL, ymax = UL), alpha = 0.2) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(limits = c(0,1), labels = percent_format(accuracy = 1)) +
  theme(
    plot.title = element_text(face = 'bold')
  ) +
  labs(
    x = 'Eval',
    y = 'Probability',
    title = abbrev('Predicted Probabilities for Membership in Evaluative Concept Class
compared to a Non-evaluative Class based on Eval', width = 80)
  )
p
ggsave(p, filename = '../output/paper/paper_study2_EvalvsNonEval_logit.pdf', width = 8,
height = 5)

p <- ggplot(newData, aes(x = Eval.Weight, y = PredictedProb)) +

```



```

geom_ribbon(aes(ymin = LL, ymax = UL), alpha = 0.2) +
geom_line() +
theme_bw() +
scale_y_continuous(limits = c(0,1), labels = percent_format(accuracy = 1)) +
theme(
  plot.title = element_text(face = 'bold')
) +
labs(
  x = 'Eval',
  y = 'Probability')
p
ggsave(p, filename =
'./output/paper/PRESI_paper_study2_EvalvsNonEval_logit_wo_title.pdf', width = 8, height
= 4)

```

```

## -----
## Evaluation metric for the logit model
response <- predict(m2, newdata = kdata, type = "response")
response <- factor(ifelse(response < .5, 'non-eval', 'eval'), levels = c('non-eval', 'eval'))
cm <- confusionMatrix(response, reference = kdata$evalType)
kdata$Adjective[!response == kdata$evalType]
kdata$evalType[!response == kdata$evalType]
cm$table
tab <- matrix(cm$overall)
rownames(tab) <- names(cm$overall)
rws <- seq(1, (nrow(tab)-1), by = 2)
col <- rep("\rowcolor[gray]{0.95}", length(rws))
print(xtable(tab, digits = 4,
  label = "tab:evalMetricsMultinom",
  caption = "Model evaluation metrics by concept class for multinomial logistic
regression."),
  comment = F,
  add.to.row = list(pos = as.list(rws), command = col),
  size = "\footnotesize", caption.placement = "top")
cm$overall

```

```

## -----
## Predict class of new elements
newData <- tibble(
  Adjective = c(
    'afraid',
    'conservative',
    'constitutional',
    'dirty',
    'happy',
    'lewd',
    'legal',

```

```

    'liberal',
    'religious'
  ),
  Eval.Weight = c(0.71, 0.45, 0.02, 0.42, 2.26, 0.25, 0.04, 0.27, 0.19)
)

```

```

response <- cbind(newData, predict(m2, newData, type = 'response', se = T))
response <- response[,-ncol(response)]
colnames(response) <- c('Adjective', 'Eval', 'Prob. for Evaluative Class', 'Standard Error')
rws <- seq(1, (nrow(response)-1), by = 2)
col <- rep("\rowcolor[gray]{0.95}", length(rws))
print(xtable(response, digits = 4,
              label = "tab:newPred",
              caption = "Predicted probabilities for formerly unobserved adjectives. If the
predicted probability is above 0.5, the adjectives would be considered evaluative."),
      comment = F, include.rownames = F,
      add.to.row = list(pos = as.list(rws), command = col),
      size = "\footnotesize", caption.placement = "top")

```

```
## -----
```

```
## Robustness-checks for truly/really ratios
```

```
## load data
```

```
rm(list=ls())
```

```
files <- list.files('../output/00-sweep/study2/', full.names = T)
```

```
files <- files[!grepl('(s|S)hiny', files)]
```

```
length(files)==6*3
```

```
df <- lapply(files, function(x){
```

```
  tmp <- read.csv2(x, stringsAsFactors = F, sep = ',') %>% as_tibble
```

```
  tmp <- mutate(tmp, fileName = gsub('.*\\|%22|\\(|\\)|\\.csv', '', x) %>% gsub('%20',
  '_', .) %>% tolower)
```

```
  tmp <- select(tmp, -txt)
```

```
  return(tmp)
```

```
})
```

```
## -----
```

```
## we had issues with short! we will thus approximate the data
```

```
df <- do.call(rbind, df)
```

```
df <- as_tibble(df)
```

```
df <- mutate(df, mod = gsub("\\_.*", "", fileName), target = gsub('.*\\_', "", fileName))
```

```
df <- mutate(df, mod = ifelse(mod == target, 'none', mod))
```

```
## sanitize short
```

```
short <- filter(df, target == 'short')
```

```
short <- short %>%
```

```
  group_by(mod, created_utc) %>%
```

```
  summarise(n=n())
```

```
shortmean <- mean(short$n[short$mod=='none'])
```

```

shortsd <- sd(short$n[short$mod=='none'])
start <- min(short$created_utc)
end <- max(short$created_utc)
expandr <- tibble(created_utc = start:end)
expandr <- expandr[!expandr$created_utc %in% short$created_utc[short$mod=='none'],]
expandr$n <- rnorm(nrow(expandr), mean = shortmean, sd = shortsd)
expandr$mod <- 'none'
expandr$target <- 'short'
shortAdd <- expandr %>% group_by(target, mod) %>% summarise(n = round(sum(n)))

## -----
## create table
fin <- df %>%
  group_by(target, mod) %>%
  summarise(n = n())
fin <- dcast(fin, 'target~mod')
## add short
fin[fin$target == 'short', 'none'] <- (fin[fin$target == 'short', 'none'] + shortAdd$n)
fin <- mutate(fin, perc_really = really/none*100, perc_truly = truly/none*100, Eval.Weight =
perc_really+perc_truly, none = as.integer(none))
## add timeframe
time <- df %>% group_by(target) %>% summarise(t2 = min(created_utc), t1 =
max(created_utc))
time <- time %>% mutate(t2 = as.POSIXct(t2, origin='1970-01-01'), t1 = as.POSIXct(t1,
origin='1970-01-01'), t_diff = difftime(t1,t2,units = 'days'))
fin <- left_join(fin, time)
fin <- mutate(fin, t1 = as.character(t1), t2 = as.character(t2), t_diff = as.character(t_diff))
print(xtable(fin[, -c(8,9)], digits = 10), include.rownames = F)

## -----
# testing for correltion between the control condition and eval values
cdata <- read.xlsx('../input/truly+very.xlsx') %>% as_tibble(.name_repair = 'unique')
cor.test(cdata$Eval.Weight...13, cdata$Normalized.for.Very, method=c("pearson"))

```