H1_energyact emb

2023-12-18

```
#Install and load necessary packages
packages <- c("devtools","here","dplyr","tidyverse","readxl","writexl","hunspell")
for (package in packages) {
  if (!requireNamespace(package, quietly = TRUE)) {
    install.packages(package)
  }
}
lapply(packages,library, character.only=T)
```

```
## Lade nötiges Paket: usethis
```

```
## here() starts at F:/Github/Embeddings_Voting
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.3     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## Warning: Paket 'hunspell' wurde unter R Version 4.3.2 erstellt
```

```
## [[1]]
## [1] "devtools"  "usethis"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "here"      "devtools"  "usethis"   "stats"     "graphics"  "grDevices"
```

```
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "dplyr"     "here"      "devtools"  "usethis"   "stats"     "graphics"
##  [7] "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "lubridate" "forcats"   "stringr"   "purrr"     "readr"     "tidyr"
##  [7] "tibble"    "ggplot2"   "tidyverse" "dplyr"     "here"      "devtools"
## [13] "usethis"   "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [19] "methods"   "base"
##
## [[5]]
##  [1] "readxl"    "lubridate" "forcats"   "stringr"   "purrr"     "readr"
##  [7] "tidyr"     "tibble"    "ggplot2"   "tidyverse" "dplyr"     "here"
## [13] "devtools"  "usethis"   "stats"     "graphics"  "grDevices" "utils"
## [19] "datasets"  "methods"   "base"
##
## [[6]]
##  [1] "writexl"   "readxl"    "lubridate" "forcats"   "stringr"   "purrr"
##  [7] "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse" "dplyr"
## [13] "here"      "devtools"  "usethis"   "stats"     "graphics"  "grDevices"
## [19] "utils"     "datasets"  "methods"   "base"
##
## [[7]]
##  [1] "hunspell"  "writexl"   "readxl"    "lubridate" "forcats"   "stringr"
##  [7] "purrr"     "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse"
## [13] "dplyr"     "here"      "devtools"  "usethis"   "stats"     "graphics"
## [19] "grDevices" "utils"     "datasets"  "methods"   "base"
```

```
#Install and load embedR
if (!requireNamespace("embedR", quietly = TRUE)) {
    # If not installed, install it using devtools
    devtools::install_github("dwulff/embedR")
}

library("embedR")
```

```
## Welcome to embedR 0.1.0!

## For more info about the package visit https://dwulff.github.io/embedR.
```

```
#read in final dataset
energyact_fin <- read_xlsx(here::here("data","energyact_final.xlsx"))

#load embeddings as R object
embedding <- readRDS(here::here("data","embedding.rds"))
```

```
#Install and load text2vec
if (!requireNamespace("text2vec", quietly = TRUE)) {
    # If not installed, install it using devtools
    install.packages("text2vec")
}
```

```r
library("text2vec")
```

```
## Warning: Paket 'text2vec' wurde unter R Version 4.3.2 erstellt
```

```r
calculate_max_similarity <- function(embedding_study, embedding_list) {
  # Calculate cosine similarity
  similarity_matrix <- sim2(embedding_study, embedding_list, method = "cosine")
  # Get the maximum similarity score for each word
  max_similarity <- apply(similarity_matrix, 1, max)
  return(max_similarity)
}
```

```r
#Load embeddings
embedding_econ <- readRDS(here::here("data","embedding_econ.rds"))

embedding_env <- readRDS(here::here("data","embedding_env.rds"))
# Creating a dataframe from the study embedding matrix
H1_df <- as.data.frame(embedding)
rownames(H1_df) <- colnames(embedding)  # Assuming rownames are the words

# Calculate max similarities
H1_df$max_similarity_econ <- calculate_max_similarity(embedding, embedding_econ)
H1_df$max_similarity_env <- calculate_max_similarity(embedding, embedding_env)
```

```r
# Set the threshold; use different thresholds
threshold <- 0.65

# Categorization logic
H1_df$category <- ifelse(H1_df$max_similarity_econ >= threshold & H1_df$max_similarity_econ > H1_df$max_
                         ifelse(H1_df$max_similarity_env >= threshold, "environmental protection", "oth
```

```r
# Set the second threshold
threshold2 <- 0.75

# Categorization logic
H1_df$category2 <- ifelse(H1_df$max_similarity_econ >= threshold & H1_df$max_similarity_econ > H1_df$ma
                         ifelse(H1_df$max_similarity_env >= threshold2, "environmental protection", "ot
```

```r
# Set the second threshold
threshold3 <- 0.8

# Categorization logic
H1_df$category3 <- ifelse(H1_df$max_similarity_econ >= threshold & H1_df$max_similarity_econ > H1_df$ma
                         ifelse(H1_df$max_similarity_env >= threshold3, "environmental protection", "ot
```

```r
#Add the new columns to the energyact_fin df. Make sure, that the rows are in the same order
if (nrow(H1_df) == nrow(energyact_fin)) {
  # The dataframes have the same number of rows
} else {
  # The dataframes have a different number of rows
  stop("The number of rows in H1_df and energyact_fin does not match.")
}
```

```
## NULL
```

```
last_five_columns <- names(H1_df)[(ncol(H1_df)-4):ncol(H1_df)]
energyact_fin[last_five_columns] <- H1_df[last_five_columns]
```

```
#Manually categorize 20 rows to check validity of embedding categorization
set.seed(26)  # Setting a seed for reproducibility
sampled_rows <- sample_n(energyact_fin, 50)

# View the sampled rows
print(sampled_rows$word)
```

```
##  [1] "mehr auflagen hausbesitzer"       "umweltschonend"
##  [3] "solaranlagen"                     "streik"
##  [5] "photovoltaik anlagen auf dächern" "wirtschaftliche chancen"
##  [7] "strompreise steigen sehr"         "verkehr"
##  [9] "isolation"                        "elektroautos"
## [11] "neue jobs"                        "landschaft?"
## [13] "nicht genug"                      "neue ideen"
## [15] "parteiübergreifend"               "luftqualität"
## [17] "strom sparen"                     "gesellschaftswandel weniger ist"
## [19] "platzproblem material"            "machbar"
## [21] "rohstoffe richtig nützen"         "greenwashing"
## [23] "nachhaltige lebensmittel"         "überlastete strassen"
## [25] "wasserkraft zu wenig"             "heizen"
## [27] "keine verbrennermotoren mehr"     "forschung"
## [29] "negativ"                          "bleiben faktentreu"
## [31] "junge zahlen zeche"               "natur"
## [33] "zu wenig durchdacht"              "ordnungsgemässe abfallentsorgung"
## [35] "ernährung"                        "saubere quellen"
## [37] "preise steigen"                   "gebäudeisolation"
## [39] "machtmissbrauch"                  "wind-energie"
## [41] "intransparent"                    "aktuelle lage"
## [43] "zeitspanne"                       "innovation"
## [45] "klima ändert trotzdem"            "esswarenvielfalt verschwindet"
## [47] "solarparks"                       "notwendige ressourcen"
## [49] "alternative rohstoffe"            "befürworter"
```

```
sampled_rows$manual_cat <- c("other","other","other","other","other","economy","economy","other","other
```

```
# Comparison
correct_matches065 <- sum(sampled_rows$category == sampled_rows$manual_cat)

correct_matches075 <- sum(sampled_rows$category2 == sampled_rows$manual_cat)

correct_matches080 <- sum(sampled_rows$category3 == sampled_rows$manual_cat)

# Calculate the ratio
match_ratio065 <- correct_matches065 / nrow(sampled_rows)

match_ratio075 <- correct_matches075 / nrow(sampled_rows)
```

```r
match_ratio080 <- correct_matches080 / nrow(sampled_rows)

# Print the ratio
print(match_ratio065)
```

```
## [1] 0.76
```

```r
print(match_ratio075)
```

```
## [1] 0.84
```

```r
print(match_ratio080)
```

```
## [1] 0.88
```

```r
# Subset to find rows where categorizations do not match
non_matching_rows065 <- sampled_rows[sampled_rows$category != sampled_rows$manual_cat, ]
non_matching_rows075 <- sampled_rows[sampled_rows$category2 != sampled_rows$manual_cat, ]
non_matching_rows080 <- sampled_rows[sampled_rows$category3 != sampled_rows$manual_cat, ]


# Print the non-matching rows
print(non_matching_rows065[, c("word", "category", "manual_cat")])
```

```
## # A tibble: 12 x 3
##    word                    category                manual_cat
##    <chr>                   <chr>                   <chr>
##  1 umweltschonend          environmental protection other
##  2 strompreise steigen sehr other                  economy
##  3 verkehr                 economy                 other
##  4 landschaft?             environmental protection other
##  5 luftqualität            environmental protection other
##  6 greenwashing            environmental protection other
##  7 forschung               economy                 other
##  8 junge zahlen zeche      other                   economy
##  9 natur                   environmental protection other
## 10 zeitspanne              other                   environmental protection
## 11 innovation              economy                 other
## 12 notwendige ressourcen   environmental protection other
```

```r
print(non_matching_rows075[, c("word", "category2", "manual_cat")])
```

```
## # A tibble: 8 x 3
##   word                    category2               manual_cat
##   <chr>                   <chr>                   <chr>
## 1 umweltschonend          environmental protection other
## 2 strompreise steigen sehr other                  economy
## 3 verkehr                 economy                 other
## 4 forschung               economy                 other
## 5 junge zahlen zeche      other                   economy
```

```
## 6 natur                    environmental protection other
## 7 zeitspanne               other                    environmental protection
## 8 innovation               economy                  other
```

```
print(non_matching_rows080[, c("word", "category3", "manual_cat")])
```

```
## # A tibble: 6 x 3
##   word                     category3 manual_cat
##   <chr>                    <chr>     <chr>
## 1 strompreise steigen sehr other     economy
## 2 verkehr                  economy   other
## 3 forschung                economy   other
## 4 junge zahlen zeche       other     economy
## 5 zeitspanne               other     environmental protection
## 6 innovation               economy   other
```

```
#How many people that mentioned economy words (H1=1) voted for (1) or against (0) the law
# Filter rows where H1 = 1 (Economy words)
economy_words <- subset(energyact_fin, category3 == "economy")

# Create a contingency table for economy words vs intended vote
economy_vote_table <- table(economy_words$intendedVote)

#Create percentages table as absolute numbers are different
economy_vote_perc <- prop.table(economy_vote_table) * 100

# Print the table
print("Percentage Table for Economy Words (category3 = economy) and Voting:")
```

```
## [1] "Percentage Table for Economy Words (category3 = economy) and Voting:"
```

```
print(economy_vote_perc)
```

```
##
##        0        1
## 29.69762 70.30238
```

```
#How many people that mentioned environmental protection words (H1=2) voted for (1) or against (0) the

# Filter rows where H1 = 2 (Environmental protection words)
env_prot_words <- subset(energyact_fin, category3 == "environmental protection")

# Create a contingency table for environmental protection words vs intended vote
env_prot_vote_table <- table(env_prot_words$intendedVote)

#Create percentages table as absolute numbers are different
env_prot_vote_perc <- prop.table(env_prot_vote_table) * 100
# Print the table
print("Table for Environmental Protection Words (category3 = environmental protection) and Voting:")
```

```
## [1] "Table for Environmental Protection Words (category3 = environmental protection) and Voting:"
```

```r
print(env_prot_vote_perc)
```

```
##
##        0        1
## 21.03004 78.96996
```

```r
#Create category3 Factor variable
# Convert to factor
energyact_fin$fcategory3 <- factor(energyact_fin$category3, levels= c("economy","environmental protecti

# Subset for 'for' voters, including only fH1 == 0 and fH1 == 1
for_voters <- subset(energyact_fin, intendedVote == 1 & fcategory3 %in% c("economy","environmental prot

# Subset for 'against' voters, including only fH1 == 0 and fH1 == 1
against_voters <- subset(energyact_fin, intendedVote == 0 & fcategory3 %in% c("economy","environmental

# Calculate percentages for economy words among 'for' and 'against' voters
economy_words_for <- sum(for_voters$fcategory3 == "economy")
percentage_economy_for <- economy_words_for / nrow(for_voters) * 100
economy_words_against <- sum(against_voters$fcategory3 == "economy")
percentage_economy_against <- economy_words_against / nrow(against_voters) * 100

# Print the results
cat("Percentage of voters 'for' mentioning economy words:", percentage_economy_for, "%\n")
```

```
## Percentage of voters 'for' mentioning economy words: 77.96407 %
```

```r
cat("Percentage of voters 'against' mentioning economy words:", percentage_economy_against, "%\n")
```

```
## Percentage of voters 'against' mentioning economy words: 84.87654 %
```

```r
# Calculate percentages for environmental protection words among 'for' and 'against' voters
env_protection_words_for <- sum(for_voters$fcategory3 == "environmental protection")
percentage_env_protection_for <- env_protection_words_for / nrow(for_voters) * 100

env_protection_words_against <- sum(against_voters$fcategory3 == "environmental protection")
percentage_env_protection_against <- env_protection_words_against / nrow(against_voters) * 100

# Print the results
cat("Percentage of voters 'for' mentioning environmental protection words:", percentage_env_protection_

```

```
## Percentage of voters 'for' mentioning environmental protection words: 22.03593 %
```

```r
cat("Percentage of voters 'against' mentioning environmental protection words:", percentage_env_protecti
```

```
## Percentage of voters 'against' mentioning environmental protection words: 15.12346 %
```

```r
#Logistic regression
# Exclude fcategory3 = other
filtered_data <- subset(energyact_fin, fcategory3 != "other")

# Create binary variables for economy and environmental protection words
filtered_data$economy_mentioned <- as.numeric(filtered_data$fcategory3 == "economy")
filtered_data$env_prot_mentioned <- as.numeric(filtered_data$fcategory3 == "environmental protection")

# Logistic regression for economy words
logit_model_economy <- glm(intendedVote ~ economy_mentioned, family = "binomial", data = filtered_data)
summary(logit_model_economy)
```

```
##
## Call:
## glm(formula = intendedVote ~ economy_mentioned, family = "binomial",
##     data = filtered_data)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.3231     0.1608   8.231   <2e-16 ***
## economy_mentioned  -0.4614     0.1761  -2.620   0.0088 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1373.5  on 1158  degrees of freedom
## Residual deviance: 1366.2  on 1157  degrees of freedom
## AIC: 1370.2
##
## Number of Fisher Scoring iterations: 4
```

```r
#Odds Ratio
odds_ratio_economy <- exp(logit_model_economy$coefficients["economy_mentioned"])
print("Odds Ratio for economy_mentioned:")
```

```
## [1] "Odds Ratio for economy_mentioned:"
```

```r
print(odds_ratio_economy)
```

```
## economy_mentioned
##          0.630415
```

```r
# Logistic regression for environmental protection words
logit_model_env_prot <- glm(intendedVote ~ env_prot_mentioned, family = "binomial", data = filtered_data)
summary(logit_model_env_prot)
```

```
##
## Call:
## glm(formula = intendedVote ~ env_prot_mentioned, family = "binomial",
##     data = filtered_data)
```

```
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.86174    0.07192   11.98   <2e-16 ***
## env_prot_mentioned   0.46138    0.17611    2.62   0.0088 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1373.5  on 1158  degrees of freedom
## Residual deviance: 1366.2  on 1157  degrees of freedom
## AIC: 1370.2
##
## Number of Fisher Scoring iterations: 4
```

```
#Odds Ratio
odds_ratio_env <- exp(logit_model_env_prot$coefficients["env_prot_mentioned"])
print("Odds Ratio for env_prot_mentioned:")
```

```
## [1] "Odds Ratio for env_prot_mentioned:"
```

```
print(odds_ratio_env)
```

```
## env_prot_mentioned
##           1.586257
```

```
#Bonferroni correction for multiple testing
# Extract p-values
p_value_economy <- summary(logit_model_economy)$coefficients["economy_mentioned", "Pr(>|z|)"]
p_value_env_prot <- summary(logit_model_env_prot)$coefficients["env_prot_mentioned", "Pr(>|z|)"]

# Combine p-values into a vector
p_values <- c(economy = p_value_economy, env_prot = p_value_env_prot)

# Bonferroni correction
p_adjusted_bonferroni <- p.adjust(p_values, method = "bonferroni")

# Holm correction (another common method)
p_adjusted_holm <- p.adjust(p_values, method = "holm")

# Printing adjusted p-values
print("Adjusted P-Values (Bonferroni):")
```

```
## [1] "Adjusted P-Values (Bonferroni):"
```

```
print(p_adjusted_bonferroni)
```

```
##    economy   env_prot
## 0.01759653 0.01759653
```

```r
print("Adjusted P-Values (Holm):")
```

```
## [1] "Adjusted P-Values (Holm):"
```

```r
print(p_adjusted_holm)
```

```
##    economy   env_prot
## 0.01759653 0.01759653
```

```r
#Calculating model fit (McFadden's R-squared)
calculate_mcfadden_r_squared <- function(model) {
  ll_full <- logLik(model)  # Log-likelihood of the full model
  ll_null <- logLik(glm(formula = intendedVote ~ 1, family = "binomial", data = model$data))  # Log-lik
  1 - as.numeric(ll_full / ll_null)
}

# McFadden's R-squared for the economy model
r_squared_economy <- calculate_mcfadden_r_squared(logit_model_economy)
print("McFadden's R-squared for the Economy Model:")
```

```
## [1] "McFadden's R-squared for the Economy Model:"
```

```r
print(r_squared_economy)
```

```
## [1] 0.005284229
```

```r
# McFadden's R-squared for the environmental protection model
r_squared_env_prot <- calculate_mcfadden_r_squared(logit_model_env_prot)
print("McFadden's R-squared for the Environmental Protection Model:")
```

```
## [1] "McFadden's R-squared for the Environmental Protection Model:"
```

```r
print(r_squared_env_prot)
```

```
## [1] 0.005284229
```