Talking Bodies: Gendered Discourse on Body Image in Instagram Comments

A Computational Social Science Analysis

Francesca Michieletto (ID 255371)

2025-09-08

Table of contents

1 Talking Bodies: Gendered Discourse on Body Image in Instagram Comments

1.1 A Computational Social Science Analysis

```
#install.packages("quanteda")
#install.packages("tm")
#install.packages("topicmodels")
#install.packages("dplyr")
#install.packages("stm")
#install.packages("wordcloud")
#install.packages("RColorBrewer")
```

```
library(dplyr)
library(topicmodels)
library(tm)
library(quanteda)
library(magrittr)
library(tidytext)
library(igraph)
library(Matrix)
library(RSpectra)
library(cluster)
library(ggraph)
library(scales)
library(ggplot2)
library(forcats)
library(tibble)
library(stm)
library(wordcloud)
library(RColorBrewer)
library(proxy)
library(quanteda.textstats)
library(quanteda.textplots)
library(tidyverse)
library(SnowballC)
library(tidyverse)
```

1.2 Loading dataset

The dataset was imported into two subsets based on gender (male and female).

```
knitr::opts_knit$set(root.dir = "/Users/francescamichieletto/Desktop/Computational Socia
data_f <- read.csv("/Users/francescamichieletto/Desktop/Computational Social Science/PRO
data_m <- read.csv("/Users/francescamichieletto/Desktop/Computational Social Science/PRO
```

The column names of the subsets were inspected to verify their structure."

```
colnames(data_m)
```

```
[1] "comment_id" "created_at" "text"       "user_id"    "username"
```

```
colnames(data_f)
```

```
[1] "comment_id" "created_at" "text"       "user_id"    "username"
```

1.3 Creating the Corpus

Two separate corpora were then created from the male (data_m) and female (data_f) datasets, specifying the text column as the document field.

```
corpus_m <- corpus(
  data_m,
  text_field  = "text",        # column containing the text of the comments
)
corpus_f <- corpus(
  data_f,
  text_field  = "text",        # column containing the text of the comments
)
```

A summary of the two corpora was generated to inspect their size, showing 45,322 documents in the male corpus and 39,017 in the female corpus.

1.4 Tokenization and preprocessing

The corpora were tokenized and preprocessed by removing punctuation, numbers, symbols, separators, URLs, and tags. Words were separated (such as hyphenated words and HTML tags), converted to lowercase, and English stop words were removed. Stemming was also applied. Additionally, social media mentions were excluded to ensure cleaner text representations for both male and female datasets.

```
tokens_m <- corpus_m %>%
  tokens(
    remove_punct = TRUE,
    remove_numbers  = TRUE,
    remove_symbols = TRUE,
    remove_separators = TRUE,
    remove_url = TRUE,
    split_hyphens = TRUE,
    split_tags = TRUE
  ) %>%
  # retained only alphabetic tokens
  tokens_select(pattern = "[A-Za-z ]", valuetype = "regex") %>%
  tokens_tolower() %>%
  tokens_remove(stopwords("en")) %>%          # english stopwords
  tokens_wordstem(language = "en")            # english stemming
# exclude social media mentions
tokens_m <- tokens_m %>%
  tokens_remove(pattern = c("^@\\w+$"), valuetype = "regex")
```

```
tokens_f <- corpus_f %>%
  tokens(
    remove_punct = TRUE,
    remove_numbers  = TRUE,
    remove_symbols = TRUE,
    remove_separators = TRUE,
    remove_url = TRUE,
    split_hyphens = TRUE,
    split_tags = TRUE
  ) %>%
  # retained only alphabetic tokens
  tokens_select(pattern = "[A-Za-z ]", valuetype = "regex") %>%
  tokens_tolower() %>%
  tokens_remove(stopwords("en")) %>%          # english stopwords
  tokens_wordstem(language = "en")            # english stemming
# exclude social media mentions
tokens_f <- tokens_f %>%
  tokens_remove(pattern = c("^@\\w+$"), valuetype = "regex")
```

1.5 Creating DFM

Document-feature matrices (DFMs) were constructed from the tokenized texts of both male and female corpora. To reduce sparsity and focus on more representative terms, the matrices were trimmed by setting a minimum document frequency threshold of 0.0001. As a result, the male corpus was reduced from 12,030 to 1,214 terms (across 45,322 documents), and the female corpus from 8,756 to 1,234 terms (across 39,017 documents).

```
dfm_m <-  dfm(tokens_m)
dfm_m <-  dfm_trim ( dfm_m ,
                          min_docfreq = 0.0001,
                          docfreq_type = "prop",
                          verbose = TRUE )
```

```
dfm_trim() changed from 12,030 features (45,322 documents) to 1,214 features (45,322 doc
```

```
dfm_f <-  dfm(tokens_f)
dfm_f <-  dfm_trim ( dfm_f ,
                          min_docfreq = 0.0001,
                          docfreq_type = "prop",
                          verbose = TRUE )
```

```
dfm_trim() changed from 8,756 features (39,017 documents) to 1,234 features (39,017 docu
```

1.6 Semantic Graph

Based on the DFMs, a semantic co-occurrence graph was constructed to capture the relationships between terms. In this representation, nodes correspond to words and edges indicate their co-occurrence within the same documents, with the weight of each edge determined by frequency. This graph structure enables the analysis of term centrality and the identification of clusters of semantically related words. First, the Semantic Graph for male corpus:

```
min_occurencies <- 5

# keep only terms with total frequency 5 across the corpus
term_freq_m <- Matrix::colSums(dfm_m)
keep_terms_m <- names(term_freq_m)[term_freq_m >= min_occurencies]
dfm_m_f <- dfm_m[, keep_terms_m]

# build the term co-occurrence matrix
A_m <- Matrix::t(dfm_m_f) %*% dfm_m_f
diag(A_m) <- 0
# each cell (i,j) represents the number of documents where terms i and j co-occur


# create an undirected weighted graph from the co-occurrence matrix
main_word_graph_m <- igraph::graph_from_adjacency_matrix(
  A_m, mode = "undirected", weighted = TRUE, diag = FALSE
)

# remove isolated nodes with degree=0
iso_m <- which(igraph::degree(main_word_graph_m) == 0)
if (length(iso_m) > 0) main_word_graph_m <- igraph::delete_vertices(main_word_graph_m, i
```

The main centrality measures, including degree, betweenness, and closeness, were computed on the semantic graph.

```
# number of nodes in the graph
cat("Number of nodes:", igraph::gorder(main_word_graph_m), "\n")
```

```
Number of nodes: 1191
```

```
# centrality measures
comp_m <- igraph::components(main_word_graph_m)
giant_m <- igraph::induced_subgraph(
  main_word_graph_m,
  vids = which(comp_m$membership == which.max(comp_m$csize))
```

```
)

degree_centrality_m      <- igraph::degree(giant_m) / (igraph::gorder(giant_m) - 1)  # c
betweeneness_centrality_m <- igraph::betweenness(giant_m, directed = FALSE, normalized =
closeness_centrality_m   <- igraph::closeness(giant_m, normalized = TRUE)

centrality_m <- list(betweeneness_centrality_m, closeness_centrality_m, degree_centralit
# list of centrality measures

head(sort(degree_centrality_m, decreasing = TRUE), 10)
```

```
      can       man        la       hai     pleas       mai     happi      para
0.2310924 0.1966387 0.1890756 0.1882353 0.1806723 0.1672269 0.1663866 0.1613445
      day      back
0.1588235 0.1579832
```

```
head(sort(betweeneness_centrality_m, decreasing = TRUE), 10)
```

```
       can         la         ya      thank         na        mai        man
0.05274783 0.04489666 0.03282943 0.03051726 0.02810227 0.02780246 0.02766326
        da        hai      pleas
0.02635086 0.02497637 0.02484677
```

```
head(sort(closeness_centrality_m, decreasing = TRUE), 10)
```

```
      can      real        ya       day       man        na       say     final
0.4652072 0.4541985 0.4538520 0.4483798 0.4475367 0.4466967 0.4433681 0.4428731
      hai      make
0.4422148 0.4407407
```

To further analyze the structure of the semantic graph, a series of graph-based functions were implemented to filter nodes, identify the most central terms, extend them with strongly connected neighbors, and perform clustering. This procedure allows for a deeper exploration of the graph's density, connectivity, and community structure.

```
min_node_degree <- 4
top_n_centrality_word <- 30

# filter nodes by minimum degree
keep_minimum_degree <- function(g, k) {
  vids <- V(g)[degree(g) >= k]
```

```
  induced_subgraph(g, vids = vids)
}

# extract the union of top-n words across different centrality measures
get_set_top_centrality_words <- function(centr_list, top = 30) {
  tops <- lapply(centr_list, function(v) names(sort(v, decreasing = TRUE))[seq_len(min(t
  unique(unlist(tops))
}

# extend the list of top words by adding strongly connected neighbor
extend_top_word <- function(g, top_words, threshold = 60) {
  if (ecount(g) == 0) return(unique(top_words))
  w <- E(g)$weight
  thr_val <- if (threshold <= 100) as.numeric(quantile(w, probs = threshold/100)) else t
  add <- character(0)
  for (tw in intersect(top_words, V(g)$name)) {
    inc_e <- E(g)[.inc(V(g)[tw])]
    for (e in inc_e) {
      if (E(g)[e]$weight >= thr_val) {
        vpair <- ends(g, e)
        other <- ifelse(vpair[1] == tw, vpair[2], vpair[1])
        add <- c(add, other)
      }
    }
  }
  unique(c(top_words, add))
}
# neighbors are added if their edge weight is above a given threshold (>100)

# keep only the giant component (largest connected subgraph)
keep_connected_components <- function(g, min_degree = NULL) {
  comp <- components(g)
  giant <- induced_subgraph(g, vids = which(comp$membership == which.max(comp$csize)))
  if (!is.null(min_degree)) giant <- keep_minimum_degree(giant, min_degree)
  giant
}

# Compute network density and average degree
network_density_and_avgdeg <- function(g) {
  n <- gorder(g); m <- gsize(g)
  dens <- edge_density(g, loops = FALSE)
  avg_deg_alt <- m / (n * (n - 1))
  list(density = dens, average_degree = avg_deg_alt)
}
```

The analysis is applied to the male semantic graph.

```
# Filter the main graph by minimum node degree
graph_min_degree_m <- keep_minimum_degree(main_word_graph_m, min_node_degree)

# select the most central words
top_words_min_m <- get_set_top_centrality_words(
  centr_list = list(betweeneness_centrality_m, degree_centrality_m, closeness_centrality
  top = top_n_centrality_word
)

# edges with weights 60th percentile are considered strong connections
top_words_extended_m <- extend_top_word(graph_min_degree_m, top_words_min_m, threshold =

# keep only the largest connected component (giant) + stricter minimum degree filter (mi
graph_cc_m <- keep_connected_components(graph_min_degree_m, min_degree = 10)

cat("number of words", length(top_words_extended_m), "\n")
```

```
number of words 1054
```

```
# compute density and average degree of the network
met_m <- network_density_and_avgdeg(graph_cc_m)
cat(sprintf("Density network %.3f\n", met_m$density))
```

```
Density network 0.041
```

```
cat(sprintf("Average Degree %.3f\n", met_m$average_degree))
```

```
Average Degree 0.020
```

The male semantic graph consists of 1,054 words, with a low density (0.041) and a very small average degree (0.020), indicating a sparse network. This suggests that word-to-word connections are not meaningful in this context.

The same analysis is applied to the female corpus.

```
min_occurencies <- 5

# keep only terms that occur>=5 times in the corpus
term_freq_f <- Matrix::colSums(dfm_f)
keep_terms_f <- names(term_freq_f)[term_freq_f >= min_occurencies]
```

```r
dfm_f_f <- dfm_f[, keep_terms_f]

# build the term co-occurrence matrix
A_f <- Matrix::t(dfm_f_f) %*% dfm_f_f
diag(A_f) <- 0

# undirected weighted graph from the co-occurrence matrix
main_word_graph_f <- igraph::graph_from_adjacency_matrix(
  A_f, mode = "undirected", weighted = TRUE, diag = FALSE
)

# remove isolated nodes
iso_f <- which(igraph::degree(main_word_graph_f) == 0)
if (length(iso_f) > 0) main_word_graph_f <- igraph::delete_vertices(main_word_graph_f, i

cat("Number of nodes:", igraph::gorder(main_word_graph_f), "\n")
```

```
Number of nodes: 990
```

```r
# extract the giant component (largest connected subgraph)
comp_f <- igraph::components(main_word_graph_f)
giant_f <- igraph::induced_subgraph(
  main_word_graph_f,
  vids = which(comp_f$membership == which.max(comp_f$csize))
)

# compute centrality measures
degree_centrality_f       <- igraph::degree(giant_f) / (igraph::gorder(giant_f) - 1)
betweeneness_centrality_f <- igraph::betweenness(giant_f, directed = FALSE, normalized =
closeness_centrality_f    <- igraph::closeness(giant_f, normalized = TRUE)

centrality_d <- list(betweeneness_centrality_f, degree_centrality_f, closeness_centralit

head(sort(degree_centrality_m, decreasing = TRUE), 10)
```

```
      can       man        la       hai     pleas       mai     happi      para
0.2310924 0.1966387 0.1890756 0.1882353 0.1806723 0.1672269 0.1663866 0.1613445
      day      back
0.1588235 0.1579832
```

```r
head(sort(betweeneness_centrality_m, decreasing = TRUE), 10)
```

```
       can         la         ya      thank         na        mai        man
0.05274783 0.04489666 0.03282943 0.03051726 0.02810227 0.02780246 0.02766326
        da        hai      pleas
0.02635086 0.02497637 0.02484677
```

```
head(sort(closeness_centrality_m, decreasing = TRUE), 10)
```

```
      can      real        ya       day       man        na       say     final
0.4652072 0.4541985 0.4538520 0.4483798 0.4475367 0.4466967 0.4433681 0.4428731
      hai      make
0.4422148 0.4407407
```

```
min_node_degree <- 4
top_n_centrality_word <- 30

graph_min_degree_f <- keep_minimum_degree(main_word_graph_f, min_node_degree)

top_words_min_f <- get_set_top_centrality_words(
  centr_list = list(betweeneness_centrality_f, degree_centrality_f, closeness_centrality
  top = top_n_centrality_word
)

top_words_extended_f <- extend_top_word(graph_min_degree_f, top_words_min_f, threshold =

graph_cc_f <- keep_connected_components(graph_min_degree_f, min_degree = 10)

cat("number of words ", length(top_words_extended_f), "\n")
```

```
number of words  871
```

```
met_f <- network_density_and_avgdeg(graph_cc_f)
cat(sprintf("Density network %.3f\n", met_f$density))
```

```
Density network 0.057
```

```
cat(sprintf("Average Degree %.3f\n", met_f$average_degree))
```

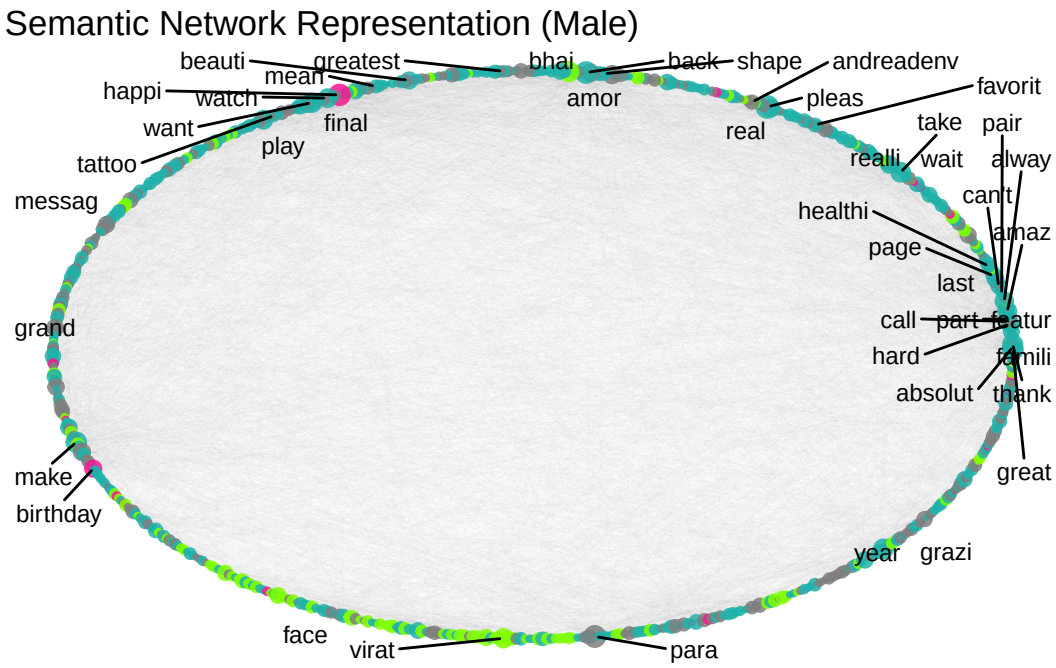
```
Average Degree 0.029
```

The female semantic graph includes 871 words, with low density (0.057) and average degree (0.029), confirming a sparse network.

The Louvain community detection algorithm was applied to the male semantic graph to identify clusters of words, which were then visualized in a circular layout, highlighting the most central terms and their community structure.

```
g_full <- graph_cc_m
# compute degree centrality for each node
V(g_full)$deg  <- as.numeric(igraph::degree(g_full))
# detect communities with Louvain algorithm
comm_obj       <- igraph::cluster_louvain(g_full, weights = E(g_full)$weight)
# assign community membership to nodes
V(g_full)$comm <- igraph::membership(comm_obj)

# Labels from top extended words
labs_pool <- intersect(top_words_extended_m, V(g_full)$name)
if (length(labs_pool) == 0) labs_pool <- V(g_full)$name

# Filter nodes with very short names (less than 4 characters)
g_full <- igraph::induced_subgraph(g_full, vids = V(g_full)[nchar(name) >= 4])

# Select the most important labels by degree (top N terms)
N_LABS <- 60
labels_set <- head(labs_pool[order(-V(g_full)$deg[labs_pool])], N_LABS)
V(g_full)$lab <- ifelse(V(g_full)$name %in% labels_set, V(g_full)$name, "")

g_plot <- g_full
# Order nodes by community and degree
ord <- order(V(g_plot)$comm, -V(g_plot)$deg)
g_plot <- igraph::permute(g_plot, ord)

# Define color palette for communities
pal3 <- c("lightseagreen","chartreuse","deeppink")
k_unique <- length(unique(V(g_plot)$comm))
pal <- c(pal3, rep("grey50", max(0, k_unique - length(pal3))))[1:k_unique]

# circular plot of the semantic network
p_disk <- ggraph::ggraph(g_plot, layout = "circle") +
  ggraph::geom_edge_link(alpha = 0.02, linewidth = 0.10, colour = "grey70") +
  ggraph::geom_node_point(aes(size = deg, colour = factor(comm)),
                          alpha = 0.85, show.legend = FALSE) +
  ggraph::geom_node_text(aes(label = lab), size = 3, repel = TRUE, max.overlaps = Inf) +
  scale_size(range = c(0.3, 3.5)) +
  scale_colour_manual(values = pal) +
```

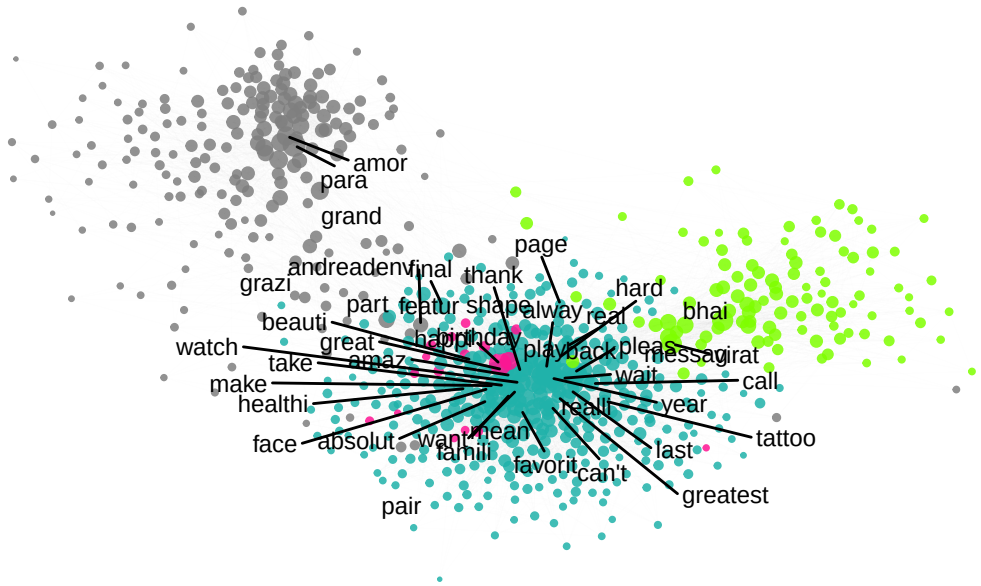
```
  theme_void() + ggtitle("Semantic Network Representation (Male)")

print(p_disk)
```



In addition to the circular layout, the semantic graph was visualized using a force-directed Fruchterman–Reingold algorithm. This layout positions nodes according to the strength of their connections, enabling a clearer view of clusters and central nodes within the network.

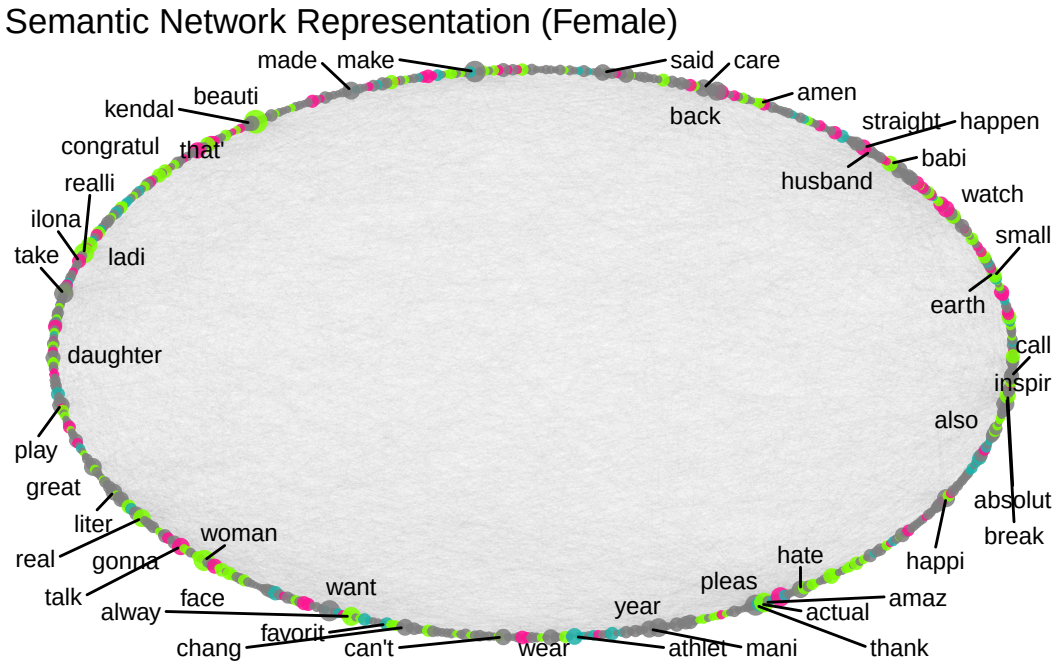
```
p_fr <- ggraph::ggraph(g_plot, layout = "fr") +
  ggraph::geom_edge_link(alpha = 0.012, linewidth = 0.10, colour = "grey70") +
  ggraph::geom_node_point(aes(size = deg, colour = factor(comm)), alpha = 0.85, show.leg
  ggraph::geom_node_text(aes(label = lab), size = 3, repel = TRUE, max.overlaps = Inf) +
  scale_size(range = c(0.3, 3.5)) +
  scale_colour_manual(values = pal) +
  theme_void() + ggtitle("Semantic Network Representation - Male")
print(p_fr)
```

Semantic Network Representation − Male

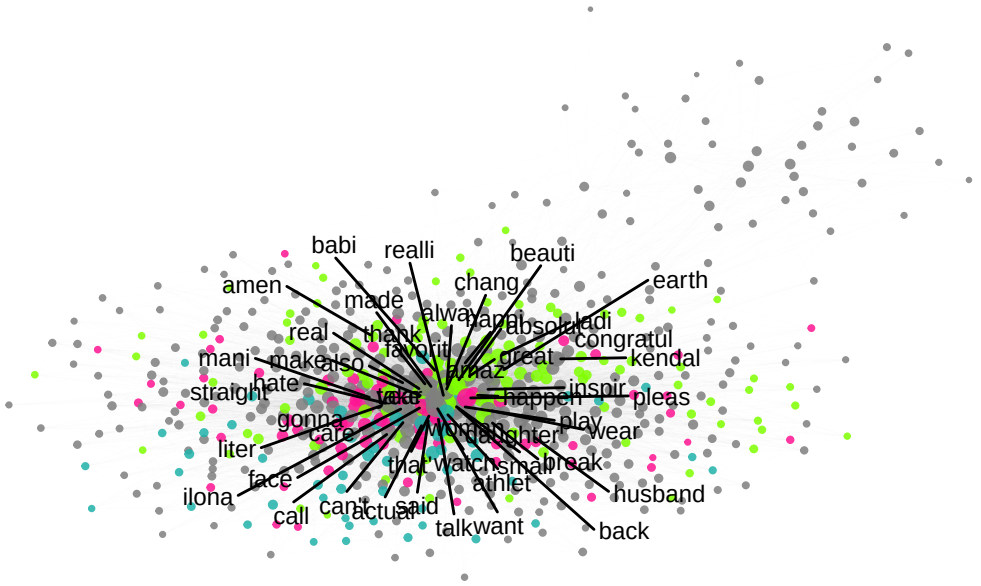



```
g_full_f <- graph_cc_f
# compute degree centrality for each node
V(g_full_f)$deg  <- as.numeric(igraph::degree(g_full_f))
# detect communities with Louvain algorithm
comm_f_obj       <- igraph::cluster_louvain(g_full_f, weights = E(g_full_f)$weight)
# assign community membership to nodes
V(g_full_f)$comm <- igraph::membership(comm_f_obj)

# Labels from top extended words
labs_pool_f   <- intersect(top_words_extended_f, V(g_full_f)$name)
if (length(labs_pool_f) == 0) labs_pool_f <- V(g_full_f)$name

# Filter nodes with very short names (less than 4 characters)
g_full_f <- igraph::induced_subgraph(g_full_f, vids = V(g_full_f)[nchar(name) >= 4])

# Select the most important labels by degree (top N terms)
N_LABS <- 60
labels_set_f <- head(labs_pool_f[order(-V(g_full_f)$deg[labs_pool_f])], N_LABS)
V(g_full_f)$lab <- ifelse(V(g_full_f)$name %in% labels_set_f, V(g_full_f)$name, "")

# Order nodes by community and degree
ord_f   <- order(V(g_full_f)$comm, -V(g_full_f)$deg)
g_plot_f <- igraph::permute(g_full_f, ord_f)

# Define color palette for communities
pal3 <- c("lightseagreen","chartreuse","deeppink")
k_unique_f <- length(unique(V(g_plot_f)$comm))
```

```
pal <- c(pal3, rep("grey50", max(0, k_unique_f - length(pal3))))[1:k_unique_f]

# circular plot of the semantic network
p_disk_f <- ggraph::ggraph(g_plot_f, layout = "circle") +
  ggraph::geom_edge_link(alpha = 0.02, linewidth = 0.10, colour = "grey70") +
  ggraph::geom_node_point(aes(size = deg, colour = factor(comm)), alpha = 0.85, show.leg
  ggraph::geom_node_text(aes(label = lab), size = 3, repel = TRUE, max.overlaps = Inf) +
  scale_size(range = c(0.3, 3.5)) +
  scale_colour_manual(values = pal) +
  theme_void() + ggtitle("Semantic Network Representation (Female)")

print(p_disk_f)
```



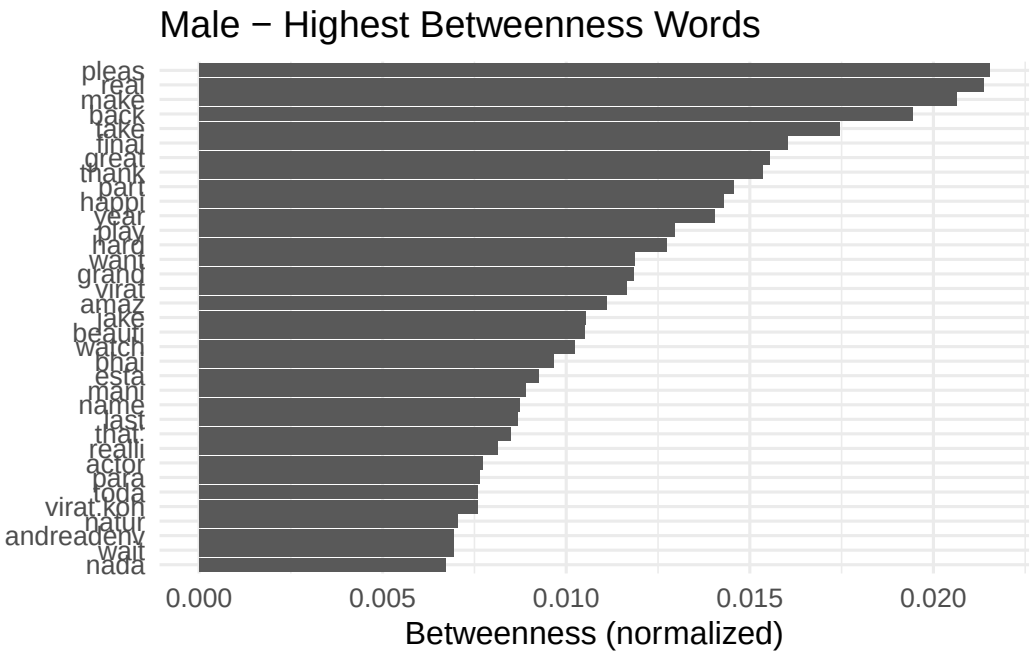

```
p_fr_f <- ggraph::ggraph(g_plot_f, layout = "fr") +
  ggraph::geom_edge_link(alpha = 0.012, linewidth = 0.10, colour = "grey70") +
  ggraph::geom_node_point(aes(size = deg, colour = factor(comm)), alpha = 0.85, show.leg
  ggraph::geom_node_text(aes(label = lab), size = 3, repel = TRUE, max.overlaps = Inf) +
  scale_size(range = c(0.3, 3.5)) +
  scale_colour_manual(values = pal) +
  theme_void() + ggtitle("Semantic Network Representation - Female")
print(p_fr_f)
```



To identify the most influential words in the male semantic network, betweenness centrality was computed. This measure highlights terms that act as bridges between different parts of the graph. The top 35 words with the highest betweenness scores (filtered to exclude very short tokens with fewer than four characters) were visualized using a bar plot.

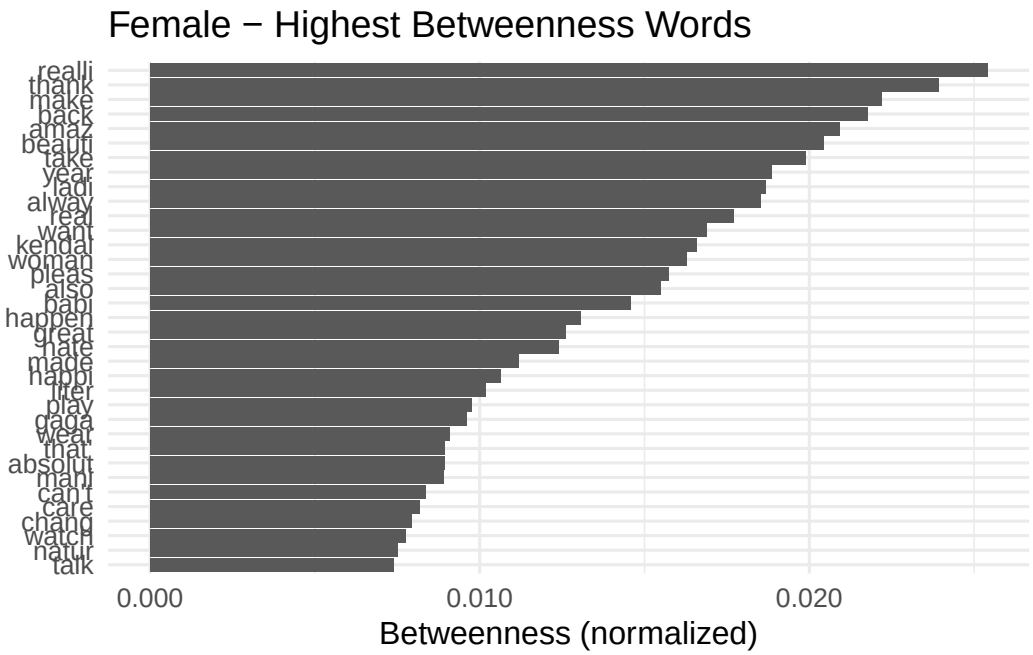
```
# top betweenness
TOP_N <- 35

# Compute betweenness centrality for each node in the male semantic graph
bet_m <- igraph::betweenness(graph_cc_m, directed = FALSE, normalized = TRUE)

# Filter out terms with less than 4 characters
bet_m <- bet_m[nchar(names(bet_m)) >= 4]
# Select the top-N words by betweenness score
top_bet_m <- sort(bet_m, decreasing = TRUE)[1:min(TOP_N, length(bet_m))]

df_bet_m <- tibble(
  term = names(top_bet_m),
  betweenness = as.numeric(top_bet_m)
) %>%
  mutate(term = fct_reorder(term, betweenness))

# horizontal bar chart
p_bet_m <- ggplot(df_bet_m, aes(x = betweenness, y = term)) +
  geom_col() +
  scale_x_continuous(labels = scales::number_format(accuracy = 0.001)) +
  labs(title = "Male - Highest Betweenness Words",
       x = "Betweenness (normalized)", y = NULL) +
```

```
  theme_minimal(base_size = 12)
print(p_bet_m)
```




```
# top betweenness
TOP_N <- 35

# Compute betweenness centrality for each node in the male semantic graph
bet_f <- igraph::betweenness(graph_cc_f, directed = FALSE, normalized = TRUE)

# Filter out terms with less than 4 characters
bet_f <- bet_f[nchar(names(bet_f)) >= 4]

top_bet_f <- sort(bet_f, decreasing = TRUE)[1:min(TOP_N, length(bet_f))]

df_bet_f <- tibble(
  term = names(top_bet_f),
  betweenness = as.numeric(top_bet_f)
) %>%
  mutate(term = fct_reorder(term, betweenness))

# horizontal bar chart
p_bet_f <- ggplot(df_bet_f, aes(x = betweenness, y = term)) +
  geom_col() +
  scale_x_continuous(labels = scales::number_format(accuracy = 0.001)) +
  labs(title = "Female - Highest Betweenness Words",
       x = "Betweenness (normalized)", y = NULL) +
```

```
  theme_minimal(base_size = 12)
print(p_bet_f)
```



1.7 LDA Topic Modelling

Converting DFM for LDA.

```
dfm_dtm_m <- tidy(dfm_m)
dfm_dtm_lda_m <- dfm_dtm_m %>%
  cast_dtm(document, term, count)

dfm_dtm_f <- tidy(dfm_f)
dfm_dtm_lda_f <- dfm_dtm_f %>%
  cast_dtm(document, term, count)
dtm_m <- dfm_dtm_lda_m
dtm_f <- dfm_dtm_lda_f
```

Latent Dirichlet Allocation (LDA) topic modelling was applied separately to the male and female corpora. Using two topics as the chosen parameter (k=2), the models identify latent semantic structures in the document-term matrices. The most representative terms for each topic were extracted, and the topic distribution across documents was also obtained.

```r
# PERFORM LDA TOPIC MODELLING
num_topics <- 2

# LDA model on the male DTM
ldamodel_m <- LDA(dtm_m, k=num_topics, control = list(seed = 1234))

# LDA model on the female DTM
ldamodel_f <- LDA(dtm_f, k=num_topics, control = list(seed = 1234))

topics_m <- tidy(ldamodel_m, matrix="beta")
topics_f <- tidy(ldamodel_f, matrix="beta")

# Select the 10 most probable words per topic
top_terms_m <- topics_m %>%
  group_by(topic) %>%
  top_n(10,beta) %>%
  arrange(topic, -beta)
print(top_terms_m)
```

```
# A tibble: 20 x 3
# Groups:   topic [2]
   topic term        beta
   <int> <chr>      <dbl>
 1     1 happi    0.0509
 2     1 man      0.0341
 3     1 pleas    0.0181
 4     1 day      0.0163
 5     1 birthday 0.0162
 6     1 thank    0.0146
 7     1 la       0.0141
 8     1 back     0.0133
 9     1 can      0.0126
10     1 beauti   0.0118
11     2 birthday 0.0377
12     2 happi    0.0219
13     2 thank    0.0189
14     2 hai      0.0143
15     2 goat     0.0115
16     2 great    0.0109
17     2 virat    0.0104
18     2 fan      0.00964
19     2 can      0.00879
20     2 la       0.00860
```

```r
top_terms_f <- topics_f %>%
  group_by(topic) %>%
  top_n(10,beta) %>%
  arrange(topic, -beta)
print(top_terms_f)
```

```
# A tibble: 20 x 3
# Groups:   topic [2]
   topic term       beta
   <int> <chr>     <dbl>
 1     1 beauti   0.0568
 2     1 pleas    0.0179
 3     1 woman    0.0179
 4     1 happi    0.0161
 5     1 amaz     0.0157
 6     1 thank    0.0153
 7     1 make     0.0146
 8     1 want     0.0129
 9     1 birthday 0.0126
10     1 back     0.0122
11     2 beauti   0.0816
12     2 thank    0.0226
13     2 la       0.0168
14     2 happi    0.0156
15     2 can      0.0147
16     2 amaz     0.0144
17     2 say      0.0125
18     2 birthday 0.0111
19     2 woman    0.0105
20     2 babi     0.0101
```

The LDA modelling with two topics highlights different sets of high-probability words in the male and female corpora. In the male dataset, topics are characterized by terms such as happi, man, pleas, birthday, and virat, suggesting mixtures of affective expressions and references to sports figures. In the female dataset, the most representative terms include beauty, woman, happi, amaz, and thank, pointing more toward evaluative and emotional content. The contrast indicates that while both corpora share affective vocabulary, the male corpus includes more personal names and contextual references, whereas the female corpus emphasizes appreciation and emotional expressions.

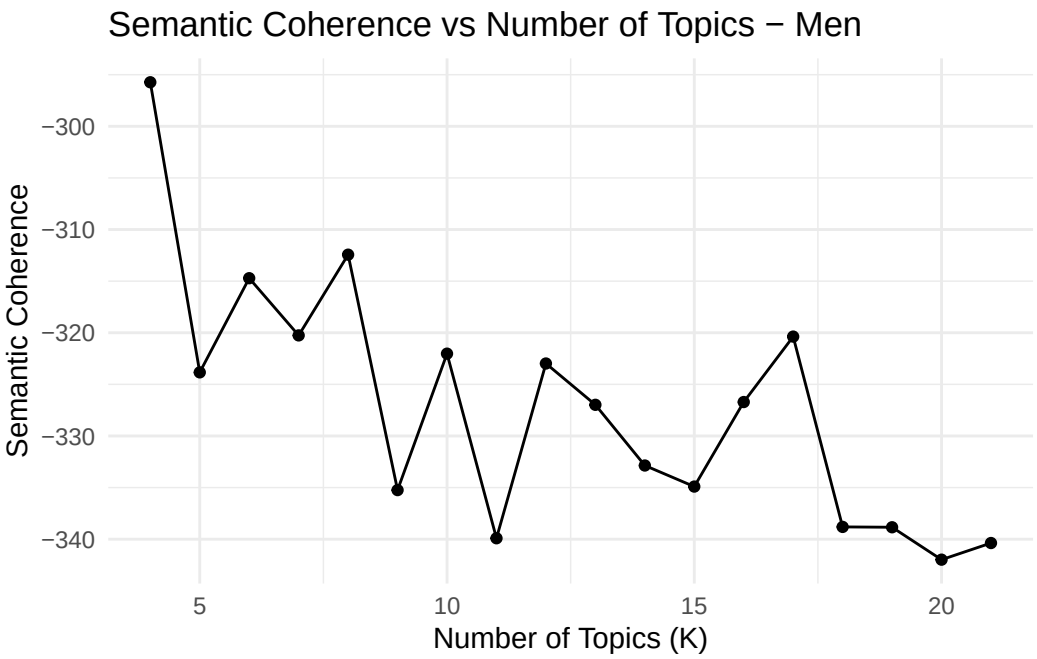
To prepare the data for Structural Topic Modelling (STM), the document-feature matrices (DFMs) were filtered to exclude empty documents and converted into STM format. A search was then performed over a range of topic numbers (K = 4–21) to evaluate model quality using semantic coherence scores for both male and female corpora.

```
# TOPIC MODELING EVALUATION
# remove empty documents (zero tokens)
dfm_m_dfm <- dfm_subset(dfm_m, ntoken(dfm_m) > 0)

# Convert DFM to STM format
dfm_stm_m <- quanteda::convert(dfm_m_dfm, to="stm")

# female
dfm_f_dfm <- dfm_subset(dfm_f, ntoken(dfm_f) > 0)
dfm_stm_f <- quanteda::convert(dfm_f_dfm, to = "stm")

# estimate models and compute diagnostics
K <- 4:21
fit_m <- searchK(dfm_stm_m$documents, dfm_stm_m$vocab, K=K)
fit_f <- searchK(dfm_stm_f$documents, dfm_stm_f$vocab, K=K)
```

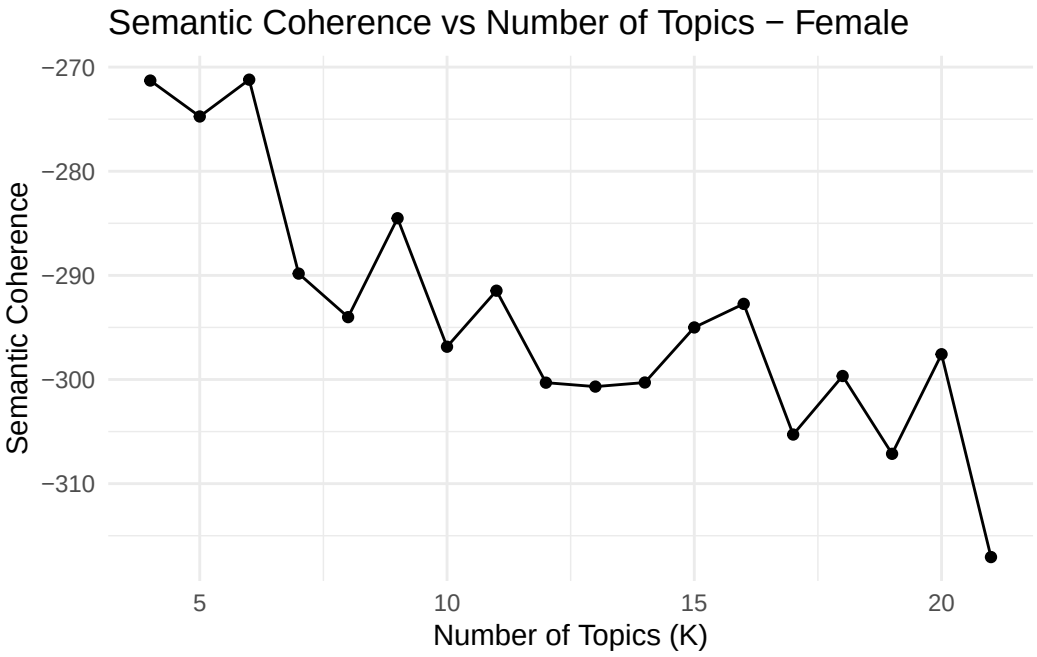
```
# extract coherence scores
plot_m <- data.frame(K=K, semanticCoherence=unlist(fit_m$results$semcoh))
plot_f <- data.frame(K=K, semanticCoherence=unlist(fit_f$results$semcoh))
```

Semantic coherence scores were computed for topic numbers ranging from 4 to 21. For the male corpus, the best coherence values appear around K=5–6, while for the female corpus the highest values are found around K=5. This suggests that relatively small topic models (around 5–6 topics) may capture the most coherent semantic structures in the data.

```
# plot semantic coherence
ggplot(plot_m, aes(x=K, y=semanticCoherence)) +
  geom_line()+
  geom_point()+
  labs(title = "Semantic Coherence vs Number of Topics - Men ",
       x = "Number of Topics (K)",
       y= "Semantic Coherence") +
  theme_minimal()
```




```
ggplot(plot_f, aes(x=K, y=semanticCoherence)) +
  geom_line()+
  geom_point()+
  labs(title = "Semantic Coherence vs Number of Topics - Female ",
       x = "Number of Topics (K)",
       y= "Semantic Coherence") +
  theme_minimal()
```



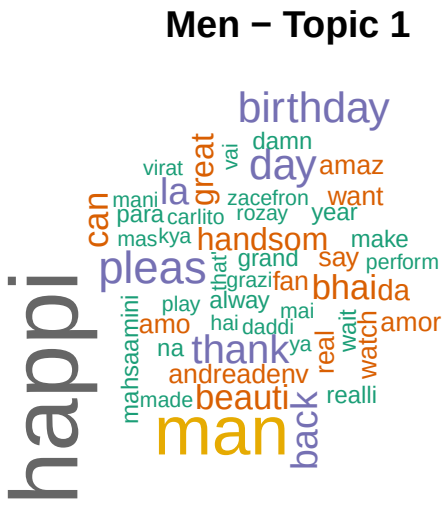
The coherence plots indicate that the highest values are reached at K=10 for male and at K=6/8 for female corpora, after which semantic coherence decreases as the number of

topics increases. This suggests that K topics provide the most interpretable and consistent representation of the data.

To visualize the topics extracted by the LDA model, word clouds were generated for both male and female corpora, displaying the 50 most probable terms per topic. In addition, bar plots were created using ggplot2 to provide a clearer quantitative view of the top terms and their associated probabilities.

```
num_topics <- 2

# worldcloud for the male corpus
par(mfrow=c(1,num_topics))
for (i in 1:num_topics) {
  terms_topic <- topics_m %>%
    filter(topic == i) %>%
    arrange(desc(beta)) %>%
    head(50)   # primi 50 termini

  wordcloud(words = terms_topic$term,
            freq = terms_topic$beta,
            max.words = 50,
            colors = brewer.pal(8, "Dark2"),
            scale = c(3,0.5))
  title(paste("Men - Topic", i))
}
```

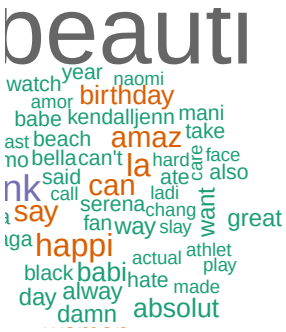
```
Warning in wordcloud(words = terms_topic$term, freq = terms_topic$beta, :
birthday could not be fit on page. It will not be plotted.
```



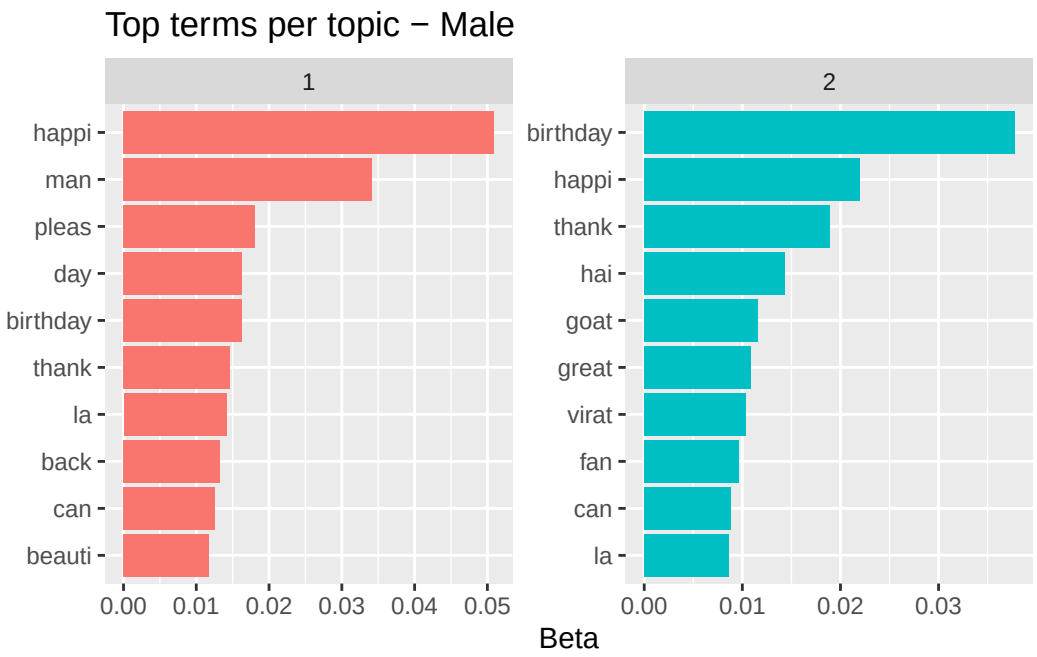
```
# worldcloud for the female corpus
par(mfrow=c(1,num_topics))
for (i in 1:num_topics) {
  terms_topic <- topics_f %>%
    filter(topic == i) %>%
    arrange(desc(beta)) %>%
    head(50)

  wordcloud(words = terms_topic$term,
            freq = terms_topic$beta,
            max.words = 50,
            colors = brewer.pal(8, "Dark2"),
            scale = c(3,0.5))
  title(paste("Female - Topic", i))
}
```

Female − Topic 1

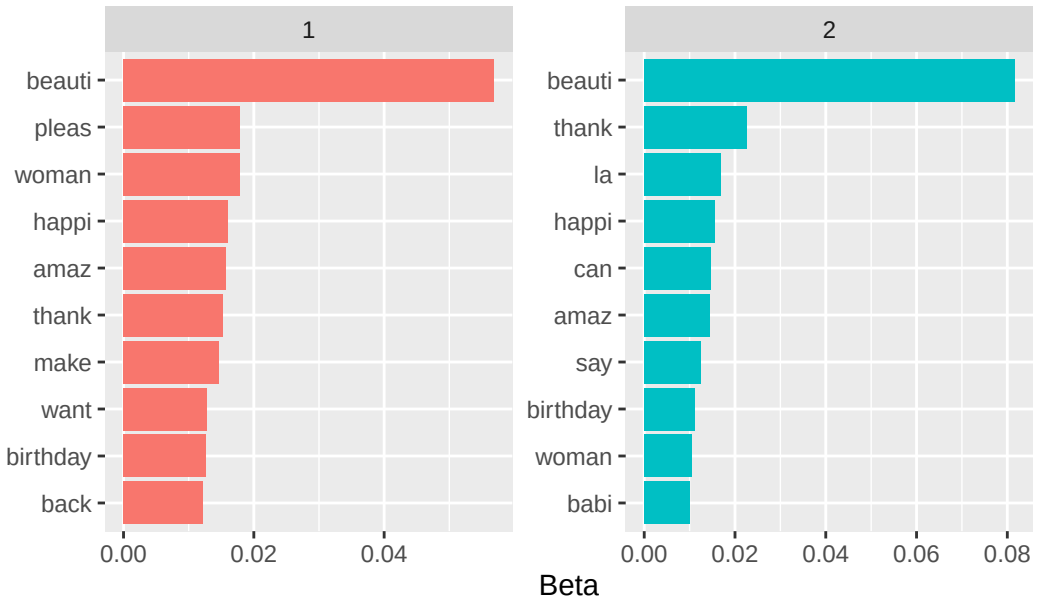
Female − Topic 2




```
# bar plots of top terms per topic
top_terms_m %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill=factor(topic))) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~ topic, scales="free") +
  coord_flip() +
  scale_x_reordered() +
  labs(title="Top terms per topic - Male", x=NULL, y="Beta")
```




```
top_terms_f %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill=factor(topic))) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~ topic, scales="free") +
  coord_flip() +
  scale_x_reordered() +
  labs(title="Top terms per topic - Female", x=NULL, y="Beta")
```



The word clouds for the male and female corpora show the most frequent terms within the two extracted LDA topics. In the male dataset, "happi," "thank," "birthday," and names such as "virat" or "ronaldo" are dominant. In the female dataset, the main terms are "beauti," "thank," "woman," "happi," and "make." These clouds highlight recurrent emotional and celebratory expressions across both groups, with differences in emphasis: sports figures and fandom references are more evident among men, while appearance-related terms (e.g., "beauti," "woman") are central in the female corpus. The barplots show the terms with the highest probability (beta) in the two topics extracted with LDA. • Male corpus: topics are dominated by terms such as happi, man, birthday, thank, reflecting celebratory content and male identity. • Female corpus: topics highlight words like beauti, pleas, woman, thank, emphasizing a lexicon more related to aesthetics, appreciation, and recognition.This section computes the cosine similarity between male and female corpora by comparing the top terms of a selected topic. The goal is to quantitatively evaluate the lexical overlap between the two groups.

```
# function to extract the top-N terms of a given topic
get_top_terms <- function(tidy_topics, topic_num, n = 20) {
  tidy_topics %>%
    filter(topic == topic_num) %>%
    arrange(desc(beta)) %>%
    slice_head(n = n)
}

# compare Topic 1 in male vs female corpora
top_m_t1 <- get_top_terms(topics_m, 1, 20)
top_f_t1  <- get_top_terms(topics_f, 1, 20)


# merge the two term sets
all_terms <- union(top_m_t1$term, top_f_t1$term)

# beta vectors
vec_m <- setNames(rep(0, length(all_terms)), all_terms)
vec_f  <- setNames(rep(0, length(all_terms)), all_terms)

vec_m[top_m_t1$term] <- top_m_t1$beta
vec_f[top_f_t1$term]   <- top_f_t1$beta

# cosine similarity between the two vectors
similarity <- simil(rbind(vec_m, vec_f), method = "cosine")
similarity
```

```
          vec_m
vec_f 0.4453913
```

```r
# compare Topic 2 in male vs female corpora
top_m_t1 <- get_top_terms(topics_m, 2, 20)
top_f_t1  <- get_top_terms(topics_f, 2, 20)


# merge the two term sets
all_terms <- union(top_m_t1$term, top_f_t1$term)

# beta vectors
vec_m <- setNames(rep(0, length(all_terms)), all_terms)
vec_f  <- setNames(rep(0, length(all_terms)), all_terms)

vec_m[top_m_t1$term] <- top_m_t1$beta
vec_f[top_f_t1$term]   <- top_f_t1$beta

# cosine similarity between the two vectors
similarity <- simil(rbind(vec_m, vec_f), method = "cosine")
similarity
```

```
          vec_m
vec_f 0.4245431
```

```r
# compare Topic 1 in male vs female corpora
top_m_t1 <- get_top_terms(topics_m, 2, 20)
top_f_t1  <- get_top_terms(topics_f, 1, 20)


# merge the two term sets
all_terms <- union(top_m_t1$term, top_f_t1$term)

# beta vectors
vec_m <- setNames(rep(0, length(all_terms)), all_terms)
vec_f  <- setNames(rep(0, length(all_terms)), all_terms)

vec_m[top_m_t1$term] <- top_m_t1$beta
vec_f[top_f_t1$term]   <- top_f_t1$beta

# cosine similarity between the two vectors
similarity <- simil(rbind(vec_m, vec_f), method = "cosine")
similarity
```

```
          vec_m
vec_f 0.4100268
```

```
# compare Topic 1 in male vs female corpora
top_m_t1 <- get_top_terms(topics_m, 1, 20)
top_f_t1  <- get_top_terms(topics_f, 2, 20)


# merge the two term sets
all_terms <- union(top_m_t1$term, top_f_t1$term)

# beta vectors
vec_m <- setNames(rep(0, length(all_terms)), all_terms)
vec_f  <- setNames(rep(0, length(all_terms)), all_terms)

vec_m[top_m_t1$term] <- top_m_t1$beta
vec_f[top_f_t1$term]   <- top_f_t1$beta

# cosine similarity between the two vectors
similarity <- simil(rbind(vec_m, vec_f), method = "cosine")
similarity
```

```
          vec_m
vec_f 0.4118305
```

The cosine similarity between male and female topics is approximately 0.41–0.44, indicating a moderate overlap in vocabulary but also clear differences in the terms emphasized by the two groups.

This step compares the vocabularies of the male and female corpora, identifying the terms that are shared across both groups as well as those that are unique to each. This allows us to quantify lexical overlap and highlight group-specific expressions.

```
# extract the vocabulary from male and female DFMs
vocab_m <- featnames(dfm_m)
vocab_f  <- featnames(dfm_f)

# shared and unique terms
common_terms  <- intersect(vocab_m, vocab_f)
unique_m <- setdiff(vocab_m, vocab_f)
unique_f  <- setdiff(vocab_f, vocab_m)

length(common_terms)
```

```
[1] 581
```

```r
length(unique_m)
```

```
[1] 633
```

```r
length(unique_f)
```

```
[1] 653
```

The vocabularies of the two corpora overlap on 581 terms, while the male corpus has 633 unique words and the female corpus 653 unique words. This indicates a balance between shared lexical material and group-specific expressions.

This section compares male and female corpora using keyness analysis to identify distinctive words for each group, and then estimates a Structural Topic Model (STM) with gender as a covariate. This allows us to analyze how topic prevalence differs between men and women.

```r
# assign group variables to DFMs
docvars(dfm_m, "group") <- "men"
docvars(dfm_f, "group") <- "female"

# merge the two DFMs
dfm_all <- rbind(dfm_m, dfm_f)

# keep only common features across both corpora
common_feats <- intersect(featnames(dfm_m), featnames(dfm_f))
dfm_all <- dfm_match(dfm_all, features = common_feats)

# identify words significantly more frequent in one group vs the other
key_f  <- textstat_keyness(dfm_all, target = docvars(dfm_all)$group == "female")
key_m <- textstat_keyness(dfm_all, target = docvars(dfm_all)$group == "men")

# top 20 distinctive terms for each group
head(key_f, 20)
```

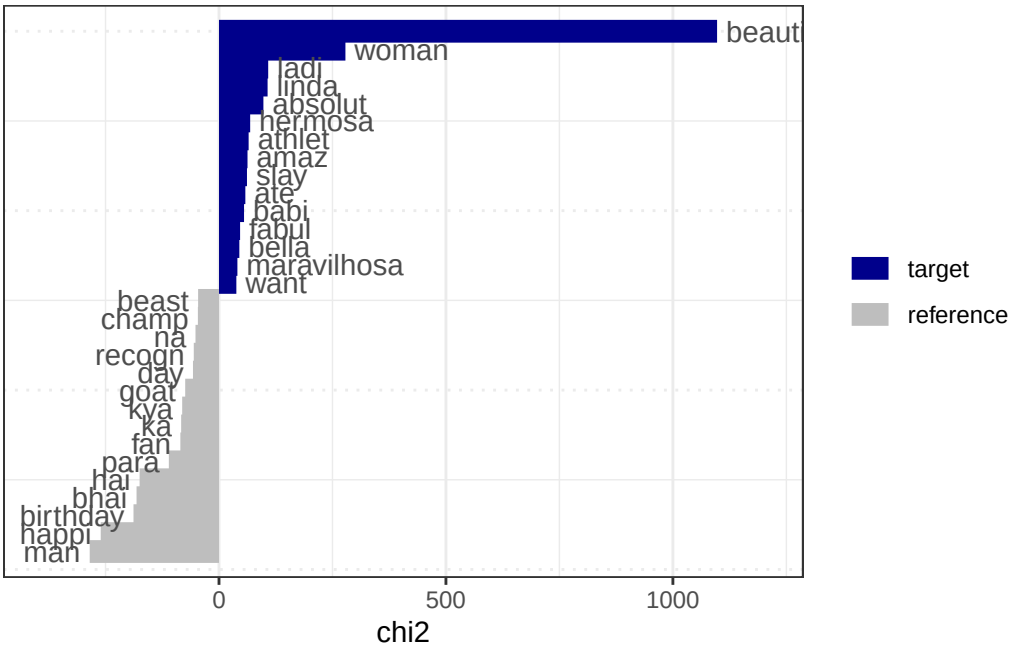
```
       feature       chi2            p n_target n_reference
1       beauti 1097.36987 0.000000e+00     1785         250
2        woman  278.51975 0.000000e+00      366          23
3         ladi  108.42852 0.000000e+00      159          15
4        linda  107.07479 0.000000e+00      160          16
5      absolut   97.83612 0.000000e+00      212          44
6      hermosa   68.67255 1.110223e-16       89           5
7       athlet   65.28927 6.661338e-16       88           6
```

```
8         amaz   62.81882 2.220446e-15      388         178
9         slay   61.65444 4.107825e-15       84           6
10         ate   58.07079 2.531308e-14       95          12
11        babi   55.14705 1.117995e-13      186          60
12       fabul   46.47677 9.271139e-12       77          10
13       bella   44.84850 2.128842e-11       82          13
14 maravilhosa   40.51587 1.950254e-10       58           5
15        want   38.27032 6.159251e-10      272         132
16       angel   37.85573 7.617432e-10       60           7
17        mama   34.90980 3.453381e-09       68          12
18        care   34.19731 4.979762e-09       97          27
19       shame   31.49537 1.999168e-08       66          13
20        also   29.69801 5.048666e-08      105          35
```

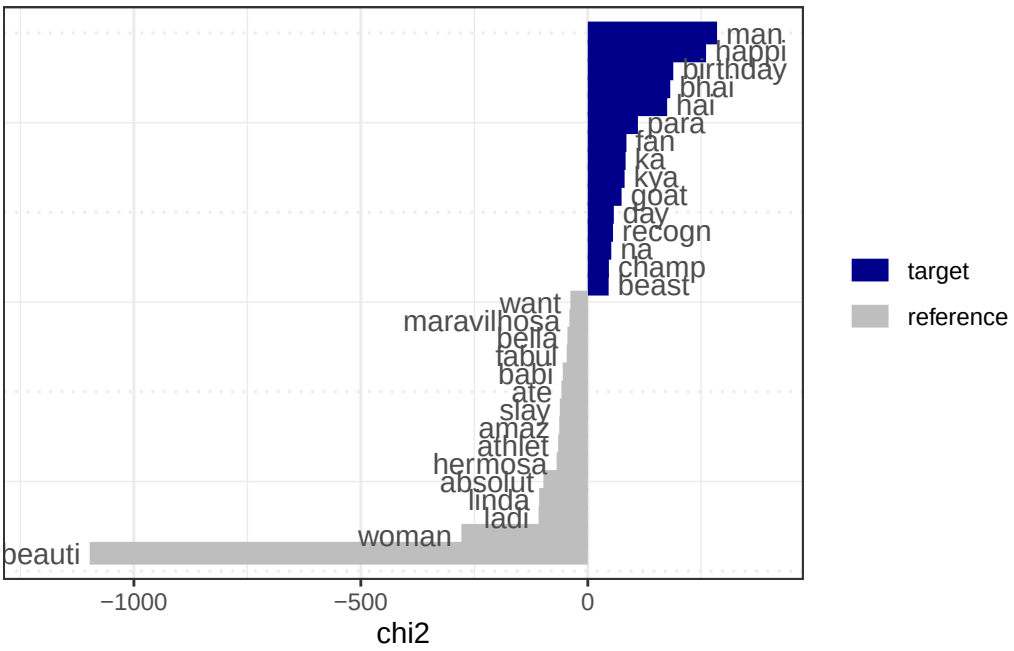
```
head(key_m, 20)
```

```
    feature      chi2            p n_target n_reference
1       man 284.83951 0.000000e+00      481         101
2     happi 260.56995 0.000000e+00      935         409
3  birthday 188.34058 0.000000e+00      693         306
4      bhai 181.67920 0.000000e+00      183           6
5       hai 174.87885 0.000000e+00      237          32
6      para 110.48439 0.000000e+00      170          30
7       fan  85.13453 0.000000e+00      213          70
8        ka  83.35389 0.000000e+00      104          11
9       kya  81.08640 0.000000e+00       95           8
10     goat  74.46337 0.000000e+00      158          44
11      day  57.35892 3.630429e-14      270         135
12   recogn  55.55497 9.092727e-14       64           5
13       na  51.75951 6.272760e-13       93          21
14    champ  46.50584 9.134582e-12       60           7
15    beast  45.98125 1.193901e-11       55           5
16    daddi  41.32401 1.289751e-10       71          15
17    grand  39.63195 3.066249e-10      111          40
18      kar  38.89416 4.474168e-10       46           4
19     vida  35.97076 2.003012e-09       76          21
20       da  35.66015 2.349205e-09      115          46
```

```
textplot_keyness(key_f, n = 15)
```




```
textplot_keyness(key_m, n = 15)
```




```
# STM WITH GENDER COVARIATE
# align vocabularies
dfm_m_matched <- dfm_match(dfm_m, features = common_feats)
dfm_f_matched <- dfm_match(dfm_f, features = common_feats)

# add group variable
docvars(dfm_m_matched, "group") <- "men"
```

```r
docvars(dfm_f_matched, "group") <- "female"

dfm_all <- rbind(dfm_m_matched, dfm_f_matched)
dfm_all <- dfm_subset(dfm_all, ntoken(dfm_all) > 0)

# STM format
out <- quanteda::convert(dfm_all, to = "stm")
out$meta$group <- factor(out$meta$group, levels = c("men","female"))  # baseline = men

# STM with K=10 topics, gender as covariate
set.seed(1234)
stm_gender <- stm(documents  = out$documents,
                  vocab      = out$vocab,
                  data       = out$meta,
                  K          = 10,
                  prevalence = ~ group,
                  init.type  = "Spectral")
```

```r
# top words for each topic
labelTopics(stm_gender, n = 10)
```

```
Topic 1 Top Words:
     Highest Prob: happi, birthday, mani, bad, hard, human, team, anyth, bath, may
     FREX: birthday, mani, bad, hard, human, team, anyth, bath, may, dad
     Lift: act, ain't, amoooo, anyth, background, bath, beau, buena, date, generat
     Score: happi, birthday, mani, bad, hard, bday, human, dad, team, bath
Topic 2 Top Words:
     Highest Prob: pleas, say, mean, call, una, beach, chang, ha, mama, wanna
     FREX: pleas, say, mean, call, una, beach, chang, ha, mama, wanna
     Lift: amigo, así, bang, channel, facebook, hace, hahaha, hang, oscar, persona
     Score: pleas, say, una, mean, beach, call, mama, chang, ha, wanna
Topic 3 Top Words:
     Highest Prob: fan, goat, bhai, that', ladi, stay, actual, vida, athlet, natur
     FREX: fan, goat, bhai, that', ladi, stay, actual, vida, athlet, natur
     Lift: aren't, aw, ay, bonita, exampl, gotta, pack, pass, sale, athlet
     Score: fan, goat, bhai, that', ladi, stay, natur, actual, recogn, vida
Topic 4 Top Words:
     Highest Prob: amaz, great, day, want, make, real, para, wait, black, hair
     FREX: amaz, great, day, want, make, real, wait, black, hair, babe
     Lift: day, make, stand, ador, asf, babe, bar, bella, black, calvin
     Score: amaz, great, want, day, make, real, para, wait, hair, black
Topic 5 Top Words:
     Highest Prob: la, babi, year, mai, said, ma, liter, care, heart, famili
     FREX: la, babi, year, said, ma, liter, care, heart, famili, mayb
```

```
     Lift: disappoint, stomach, straight, babi, famili, heart, la, mayb, saw, ago
     Score: la, babi, year, ma, said, mai, liter, care, heart, dear
Topic 6 Top Words:
     Highest Prob: alway, woman, take, dream, congratul, share, yeah, anyon, ask, name
     FREX: alway, woman, take, dream, congratul, share, yeah, anyon, ask, name
     Lift: action, admir, angl, ask, ciao, comeback, dam, earth, easi, favourit
     Score: woman, alway, take, congratul, dream, yeah, share, ask, anyon, name
Topic 7 Top Words:
     Highest Prob: thank, damn, hai, ya, talk, ka, wear, esta, kya, head
     FREX: thank, damn, hai, ya, ka, wear, esta, head, read, daddi
     Lift: ahí, appreci, away, awww, cap, compar, cuando, daddi, dalla, das
     Score: thank, hai, damn, ya, ka, esta, talk, daddi, wear, read
Topic 8 Top Words:
     Highest Prob: back, way, da, can't, made, favorit, ass, happen, okay, fabul
     FREX: back, way, da, can't, made, favorit, ass, happen, okay, fabul
     Lift: hahahaha, back, actor, ah, alreadi, american, anybodi, appear, ass, bag
     Score: back, way, da, can't, favorit, made, ass, okay, happen, fabul
Topic 9 Top Words:
     Highest Prob: beauti, man, amo, absolut, play, linda, amor, grand, awesom, face
     FREX: beauti, amo, absolut, linda, awesom, face, also, inspir, eat, ate
     Lift: ahh, ate, woah, also, linda, beauti, absolut, af, al, amiga
     Score: beauti, man, absolut, amo, linda, ate, awesom, also, face, slay
Topic 10 Top Words:
     Highest Prob: can, realli, watch, last, na, speak, agre, lmao, player, vai
     FREX: can, realli, watch, last, na, speak, agre, lmao, player, vai
     Lift: ain't, aliv, baddest, brain, breath, catch, clean, crack, creat, dia
     Score: can, realli, watch, last, na, vai, speak, agre, player, fat
```

```r
# effects of gender on topic prevalence
prep <- estimateEffect(1:10 ~ group, stmobj = stm_gender, metadata = out$meta, uncertain
summary(prep)
```

```
Call:
estimateEffect(formula = 1:10 ~ group, stmobj = stm_gender, metadata = out$meta,
    uncertainty = "Global")


Topic 1:

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.148480   0.001907   77.86   <2e-16 ***
groupfemale -0.054506   0.002546  -21.41   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Topic 2:

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0796117  0.0008225  96.793   <2e-16 ***
groupfemale -0.0018353  0.0011067  -1.658   0.0973 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Topic 3:

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.094361   0.001028   91.77   <2e-16 ***
groupfemale -0.016454   0.001356  -12.13   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Topic 4:

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.1103636  0.0009082 121.523  < 2e-16 ***
groupfemale 0.0048083  0.0012519   3.841 0.000123 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Topic 5:

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0791002  0.0006319 125.179  < 2e-16 ***
groupfemale 0.0073228  0.0009608   7.621 2.61e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Topic 6:
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0745228  0.0008007   93.08   <2e-16 ***
groupfemale 0.0191181  0.0010201   18.74   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Topic 7:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1239058  0.0009697  127.77   <2e-16 ***
groupfemale -0.0272792  0.0013063  -20.88   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Topic 8:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0908021  0.0008174 111.084  < 2e-16 ***
groupfemale -0.0074621  0.0012089  -6.173 6.83e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Topic 9:

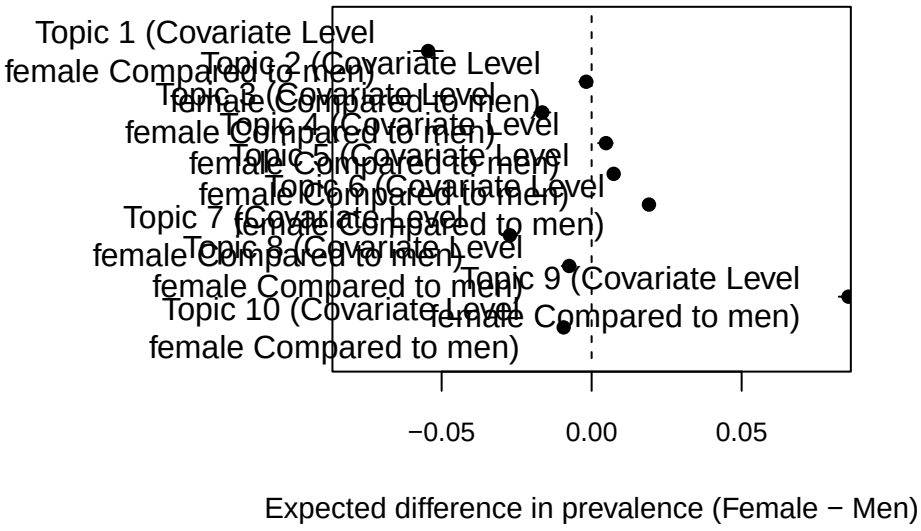
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.123638   0.001092  113.26   <2e-16 ***
groupfemale 0.085562   0.001762   48.55   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Topic 10:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0752470  0.0007315 102.872   <2e-16 ***
groupfemale -0.0093241  0.0009440  -9.878   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
op <- par(mar = c(5, 12, 3, 2))  # più spazio a sinistra
plot(prep, covariate = "group", method = "difference",
     cov.value1 = "female", cov.value2 = "men",
     topics = 1:10,
     xlab = "Expected difference in prevalence (Female - Men)",
     xlim = c(-0.08, 0.08),    # regola se serve
     cex = 0.9, cex.lab = 0.9, cex.axis = 0.8)
```




```r
par(op)
```

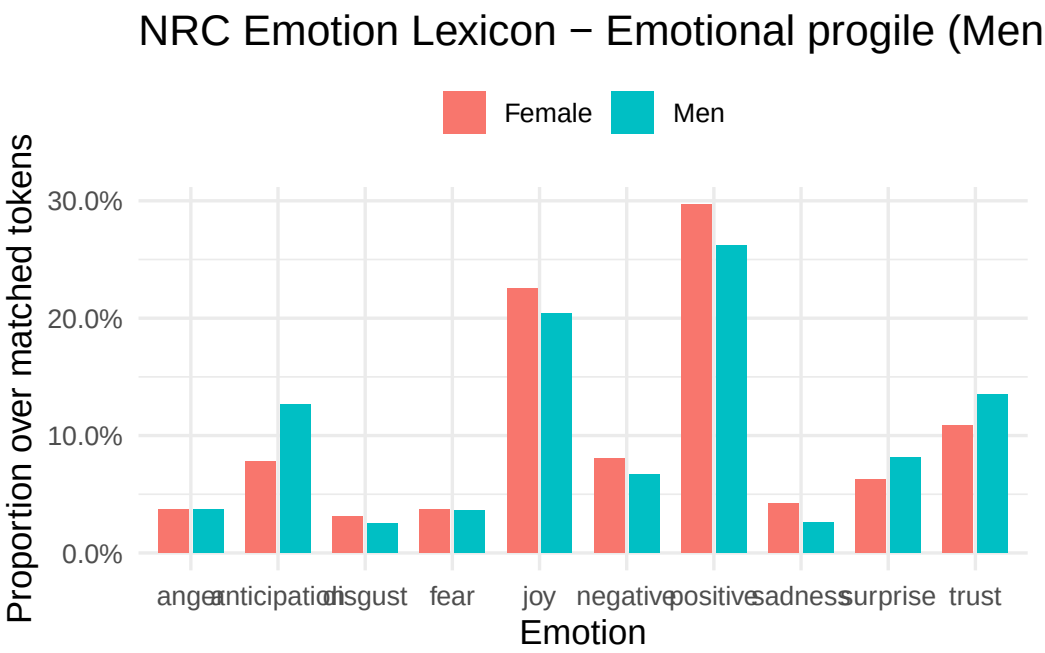
The plot shows the expected difference in topic prevalence between comments directed at women and men. Positive values indicate topics more prevalent in female comments (e.g., Topics 4, 5, 7), while negative values indicate topics more prevalent in male comments (e.g., Topics 10, 3). Some topics (e.g., 6, 8, 9) are balanced across genders.

1.8 Detect emotions

We estimate emotion profiles for male and female corpora using the NRC Emotion Lexicon. DFMs are harmonized (same features; alphabetic tokens 4 chars), the lexicon is stem-aligned to our tokens, and emotion counts are aggregated and normalized to compare proportions across groups.

```r
theme_set(theme_minimal(base_size = 13))

# DFMs coherence
dfm_m_em <- dfm_select(dfm_m, pattern = "^[[:alpha:]]{4,}$",
                       valuetype = "regex", selection = "keep")
dfm_f_em <- dfm_select(dfm_f,  pattern = "^[[:alpha:]]{4,}$",
                       valuetype = "regex", selection = "keep")

feats_union <- union(featnames(dfm_m_em), featnames(dfm_f_em))
dfm_m_em  <- dfm_match(dfm_m_em, features = feats_union)
dfm_f_em  <- dfm_match(dfm_f_em, features = feats_union)

# RC Emotion Lexicon (English) and stem
nrc <- tidytext::get_sentiments("nrc") %>%  # columns: word, sentiment
  rename(term = word, emotion = sentiment) %>%
  mutate(term = tolower(term),
         term = wordStem(term, language = "en")) %>%
  filter(nchar(term) >= 4) %>%
  distinct(term, emotion)

# quanteda dictionary: emotion -> terms
nrc_dict <- dictionary(split(nrc$term, nrc$emotion))

# dictionary lookup
emo_m_doc <- dfm_lookup(dfm_m_em, dictionary = nrc_dict)
emo_f_doc <- dfm_lookup(dfm_f_em, dictionary = nrc_dict)

# aggregate to corpus level
emo_m_tot <- colSums(emo_m_doc)
emo_f_tot <- colSums(emo_f_doc)

tok_m_map <- sum(emo_m_tot)
tok_f_map <- sum(emo_f_tot)

summary_df <- tibble(
  emotion = names(emo_m_tot),
  count_m = as.numeric(emo_m_tot),
  count_f = as.numeric(emo_f_tot)
) %>%
  mutate(
    prop_m = count_m / tok_m_map,
    prop_f = count_f / tok_f_map,
    diff_mf = prop_m - prop_f
  ) %>%
```

```r
  arrange(desc(pmax(prop_m, prop_f)))

# bars (Male vs Female)
plot_bar <- summary_df %>%
  pivot_longer(c(prop_m, prop_f), names_to = "group", values_to = "prop") %>%
  mutate(group = recode(group, prop_m = "Men", prop_f = "Female")) %>%
  ggplot(aes(x = emotion, y = prop, fill = group)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.1)) +
  labs(title = "NRC Emotion Lexicon - Emotional progile (Men vs Female)",
       x = "Emotion", y = "Proportion over matched tokens", fill = NULL) +
  theme(legend.position = "top")

print(plot_bar)
```




```r
summary_out <- summary_df %>%
  transmute(
    emotion,
    count_m, count_f,
    prop_m = scales::percent(prop_m, 0.1),
    prop_f = scales::percent(prop_f, 0.1),
    diff_mf = scales::percent(diff_mf, 0.1)
  )
summary_out
```

```
# A tibble: 10 x 6
```

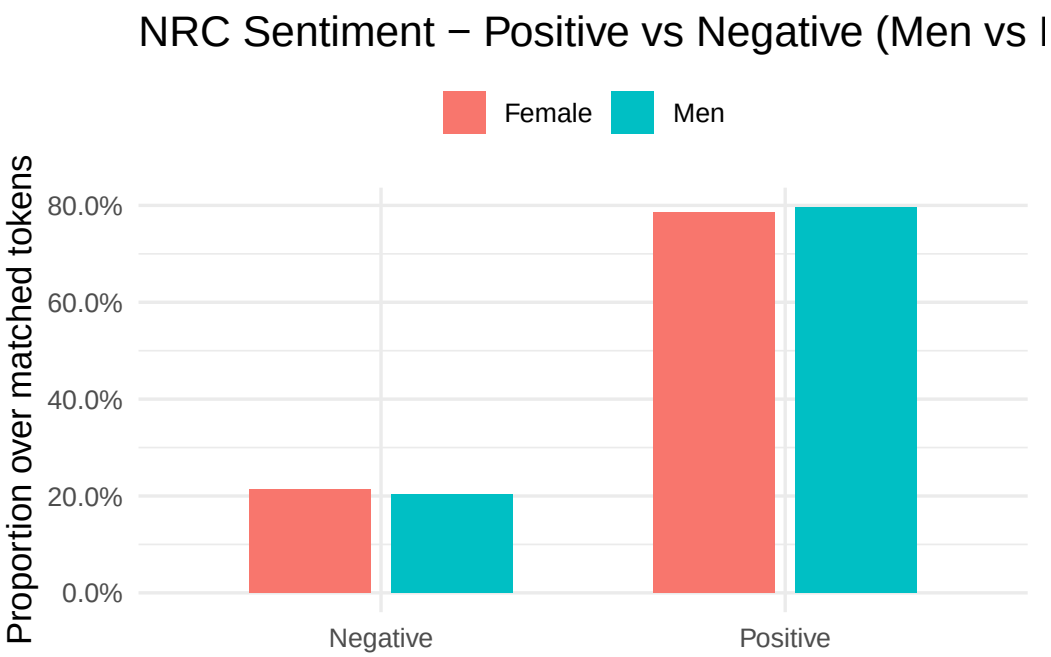
```
   emotion      count_m count_f prop_m prop_f diff_mf
   <chr>          <dbl>   <dbl> <chr>  <chr>  <chr>
 1 positive        5256    7443 26.2%  29.7%  -3.5%
 2 joy             4096    5646 20.4%  22.5%  -2.1%
 3 trust           2706    2722 13.5%  10.9%  2.6%
 4 anticipation    2545    1957 12.7%  7.8%   4.9%
 5 surprise        1627    1572 8.1%   6.3%   1.8%
 6 negative        1341    2028 6.7%   8.1%   -1.4%
 7 sadness          530    1047 2.6%   4.2%   -1.5%
 8 fear             735     940 3.7%   3.7%   -0.1%
 9 anger            751     937 3.7%   3.7%   0.0%
10 disgust          505     779 2.5%   3.1%   -0.6%
```

Both male and female comments are overwhelmingly positive, but women lean more towards joy and sadness, while men show more anticipation and surprise. The negative emotions are minor and evenly distributed.

We replicate the analysis focusing only on NRC Positive/Negative labels, report normalized proportions for each group, and quantify effect size with Cohen's h.

```r
theme_set(theme_minimal(base_size = 13))

# DFMs coherence
dfm_m_pn <- dfm_select(dfm_m, pattern = "^[[:alpha:]]{4,}$",
                       valuetype = "regex", selection = "keep")
dfm_f_pn <- dfm_select(dfm_f,  pattern = "^[[:alpha:]]{4,}$",
                       valuetype = "regex", selection = "keep")
feats_union <- union(featnames(dfm_m_pn), featnames(dfm_f_pn))
dfm_m_pn  <- dfm_match(dfm_m_pn, features = feats_union)
dfm_f_pn  <- dfm_match(dfm_f_pn, features = feats_union)

# NRC lexicon: keep only Positive/Negative
nrc_pn <- tidytext::get_sentiments("nrc") %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  transmute(
    term    = tolower(word) %>% wordStem(language = "en"),
    label   = sentiment
  ) %>%
  filter(nchar(term) >= 4) %>%
  distinct(term, label)

# build dictionary
pn_dict   <- dictionary(split(nrc_pn$term, nrc_pn$label))
pn_m_doc  <- dfm_lookup(dfm_m_pn, dictionary = pn_dict)
pn_f_doc  <- dfm_lookup(dfm_f_pn, dictionary = pn_dict)
```

```r
# aggregate to corpus level
pn_m_tot <- colSums(pn_m_doc)
pn_f_tot <- colSums(pn_f_doc)

tok_m_map_pn <- sum(pn_m_tot)
tok_f_map_pn <- sum(pn_f_tot)

pn_summary <- tibble(
  label   = names(pn_m_tot),
  count_m = as.numeric(pn_m_tot),
  count_f = as.numeric(pn_f_tot)
) %>%
  mutate(
    prop_m  = count_m / tok_m_map_pn,
    prop_f  = count_f / tok_f_map_pn,
    diff_mf = prop_m - prop_f
  )

# Cohen's h for two independent proportions
cohens_h <- function(p1, p2) 2*asin(sqrt(p1)) - 2*asin(sqrt(p2))
pn_summary <- pn_summary %>%
  mutate(cohen_h = cohens_h(prop_m, prop_f)) %>%
  arrange(desc(label))  #

# plots
plot_pn_bar <- pn_summary %>%
  pivot_longer(c(prop_m, prop_f), names_to = "group", values_to = "prop") %>%
  mutate(group = recode(group, prop_m = "Men", prop_f = "Female"),
         label = ifelse(label == "positive", "Positive", "Negative")) %>%
  ggplot(aes(x = label, y = prop, fill = group)) +
  geom_col(position = position_dodge(width = 0.7), width = 0.6) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.1)) +
  labs(title = "NRC Sentiment - Positive vs Negative (Men vs Female)",
       x = NULL, y = "Proportion over matched tokens", fill = NULL) +
  theme(legend.position = "top")

print(plot_pn_bar)
```




```r
pn_out <- pn_summary %>%
  transmute(
    label,
    count_m, count_f,
    prop_m  = scales::percent(prop_m, 0.1),
    prop_f  = scales::percent(prop_f, 0.1),
    diff_mf = scales::percent(diff_mf, 0.1),
    cohen_h = round(cohen_h, 3)
  )
pn_out
```

```
# A tibble: 2 x 7
  label    count_m count_f prop_m prop_f diff_mf cohen_h
  <chr>      <dbl>   <dbl> <chr>  <chr>  <chr>     <dbl>
1 positive    5256    7443 79.7%  78.6%  1.1%      0.027
2 negative    1341    2028 20.3%  21.4%  -1.1%    -0.027
```

Overall, the emotion and sentiment analyses reveal that comments directed at men and women share a largely similar emotional structure, with only minor differences in prevalence. While positive emotions are dominant across both groups, women receive relatively more joy and sadness, whereas men's comments contain slightly more anticipation and surprise. The polarity (positive vs. negative) is nearly identical, suggesting that gender does not drastically alter the overall tone of engagement, but subtle variations in emotional framing may still carry interpretive significance.