

Heard or Halted?

Analyzing the Emotional Tone of Judicial Interruptions

Tian Tong, Georgetown University

2025-04-20

Abstract

This script analyzes gender dynamics in U.S. Supreme Court oral arguments, extending the analysis of Cai et al. (2024). It uses 12,663 speech chunks derived from the ConvoKit Supreme Court Corpus (2010–2019), obtained from the original authors’ public replication data. It addresses two primary questions:

- 1) *whether interruptions alter the semantic content of an advocate’s argument,*
- 2) *if interruptions directed at female advocates are characterized by more negative sentiment.*

To measure semantic shift, the script uses pre-trained GloVe word vectors to create aggregate embeddings for interrupted vs. non-interrupted speech and calculates their cosine similarity. For sentiment analysis, it applies the NRC Emotion Lexicon to interrupted chunks to derive a negative sentiment ratio, which is then compared across genders using a t-test and OLS regression. Latent Dirichlet Allocation (LDA) is also explored as a robustness check, though this method did not ultimately yield an appropriate model for the final analysis.

Overall, the analysis finds that interruptions do not substantially alter the semantic substance of advocates’ arguments, but interruptions directed at female advocates contain a consistently higher proportion of negative emotional language, indicating a small yet robust gendered disparity in the tone of Supreme Court oral argument interactions.

The full reproducible workflow and report are available at: https://github.com/1TSHARUKA/Emotional_Interruption_Analysis.

Preparation

Setup

```
# Reproducibility
set.seed(42)

# Data input
df <- read.csv("data/df_final_with_text.csv")

summary(df$chunk_text)
```

```
##      Length      Class      Mode
##      12663 character character
```

The length, 12,663, matches the total number of advocate speech chunks referenced in the analysis.

Preprocessing

```
## Remove duplicate speech chunks based on unique advocate-justice interaction IDs
df <- df %>%
  distinct(case_id, justice_name, advocate_name, utt_id_first, utt_id_last, .keep_all = TRUE)

nrow(df)
```

```
## [1] 12663
```

This indicates that each row in the original dataframe was already unique based on the combination of case, justice, advocate, and utterance IDs, serving as a successful data integrity check.

```
# Keep only one unique record per speech chunk (remove repeated rows)
df_unique <- df %>%
  group_by(case_id, utt_id_first, utt_id_last) %>%
  slice(1) %>%
  ungroup()

nrow(df_unique)
```

```
## [1] 12663
```

Similar to the previous step, this output of 12663 shows that this grouping and slicing operation did not remove any rows. This further confirms the dataset is already structured with one row per unique speech chunk, as expected for the analysis.

```
# Clean and normalize text for analysis
df_cleaned <- df_unique %>%
  mutate(
    chunk_text_clean = chunk_text %>%
      replace_non_ascii() %>%      # remove weird characters
      replace_contraction() %>%    # convert "don't" -> "do not"
      tolower() %>%                # lowercase
      str_replace_all("[[:punct:]]", "") %>% # remove punctuation
      str_squish()
  )
```

```
# Inspect result
head(df_cleaned, 10)
```

```
## # A tibble: 10 x 31
##   case_id      case_year justice_name  advocate_name utt_id_first utt_id_last
##   <chr>      <int> <chr>      <chr>      <chr>      <chr>
## 1 2010_08-1314    2010 Antonin Scalia Martin N. B~ 22763__0_001 22763__0_0~
## 2 2010_08-1314    2010 John G. Robert~ Martin N. B~ 22763__0_009 22763__0_0~
## 3 2010_08-1314    2010 Anthony M. Ken~ Martin N. B~ 22763__0_021 22763__0_0~
## 4 2010_08-1314    2010 Ruth Bader Gin~ Martin N. B~ 22763__0_027 22763__0_0~
## 5 2010_08-1314    2010 Anthony M. Ken~ Martin N. B~ 22763__0_031 22763__0_0~
## 6 2010_08-1314    2010 John G. Robert~ Martin N. B~ 22763__0_035 22763__0_0~
## 7 2010_08-1314    2010 Antonin Scalia Martin N. B~ 22763__0_047 22763__0_0~
## 8 2010_08-1314    2010 Sonia Sotomayor Gregory G. G~ 22763__2_000 22763__2_0~
## 9 2010_08-1314    2010 Stephen G. Bre~ Gregory G. G~ 22763__2_004 22763__2_0~
## 10 2010_08-1314    2010 Sonia Sotomayor Gregory G. G~ 22763__2_012 22763__2_0~
## # i 25 more variables: advocate_gender <chr>, num_utts <int>,
## #   num_utts_adv <int>, num_utts_justice <int>, num_toks_total <int>,
## #   num_toks_adv <int>, num_toks_justice <int>, advocate_ideology <chr>,
```

```
## #   justice_ideology <chr>, adv_experience_int <int>, adv_experience_bin <int>,
## #   female_issue <int>, num_adv_utts_interrupted <int>,
## #   num_justice_utts_interrupted <int>, adv_interruption_rate <dbl>,
## #   justice_interruption_rate <dbl>, num_adv_disfl <int>, ...
```

Embedding with GloVe (100d)

Load GloVe 100d

We are loading the 100-dimensional (100d) GloVe word vectors. This specific model was a pre-selected dataset for this project. The 100d version is also a standard practice for semantic analysis, as it offers a robust balance between capturing detailed semantic meaning and maintaining computational efficiency (compared to larger 200d or 300d models).

```
# Load pre-trained GloVe embeddings (100-dimensional) for semantic similarity analysis
glove_path <- "data/glove.6B.100d.txt"
```

```
# Read GloVe file
glove <- fread(
  glove_path,
  header = FALSE,
  sep = " ",
  quote = "",
  encoding = "UTF-8",
  data.table = FALSE
)
```

```
# Separate word column from numeric columns
words <- glove[, 1]
vectors <- as.matrix(glove[, -1])
rownames(vectors) <- words
```

```
# Check column names
colnames(df_cleaned)
```

```
## [1] "case_id"
## [2] "case_year"
## [3] "justice_name"
## [4] "advocate_name"
## [5] "utt_id_first"
## [6] "utt_id_last"
## [7] "advocate_gender"
## [8] "num_utts"
## [9] "num_utts_adv"
## [10] "num_utts_justice"
## [11] "num_toks_total"
## [12] "num_toks_adv"
## [13] "num_toks_justice"
## [14] "advocate_ideology"
## [15] "justice_ideology"
## [16] "adv_experience_int"
## [17] "adv_experience_bin"
## [18] "female_issue"
## [19] "num_adv_utts_interrupted"
## [20] "num_justice_utts_interrupted"
```

```
## [21] "adv_interruption_rate"
## [22] "justice_interruption_rate"
## [23] "num_adv_disfl"
## [24] "num_justice_disfl"
## [25] "num_adv_toks_in_utts_interrupted"
## [26] "num_justice_toks_in_utts_interrupted"
## [27] "justice_gender"
## [28] "adv_ideology_gender"
## [29] "ideology_matches"
## [30] "chunk_text"
## [31] "chunk_text_clean"
```

Tokenize cleaned text

```
# Tokenize cleaned text and keep only words present in the GloVe vocabulary
df_tokens <- df_cleaned %>%
  unnest_tokens(word, chunk_text_clean) %>%
  filter(word %in% rownames(vectors))
```

Create document embeddings

```
# Function to compute mean GloVe embedding
get_chunk_embedding <- function(words_in_chunk) {
  mat <- vectors[words_in_chunk, , drop = FALSE]
  if (nrow(mat) == 0) return(rep(NA, 100)) # if no valid words
  colMeans(mat)
}

# Apply it by chunk (utt_id_first + utt_id_last to uniquely ID chunk)
df_embeddings <- df_tokens %>%
  group_by(case_id, utt_id_first, utt_id_last) %>%
  summarise(across(word, list), .groups = "drop") %>%
  mutate(embedding = map(word, get_chunk_embedding)) %>%
  unnest_wider(embedding, names_sep = "_dim_")
```

1. The `get_chunk_embedding` function:

- Takes a vector of word tokens for a single chunk.
- Looks up the 100d GloVe embedding for each token in the pre-loaded `vectors` matrix.
- Averages these vectors by calculating across all valid words in the chunk to create a single 100-dimensional vector representing the aggregate semantic content of the chunk.
- Returns this single 100-dimensional vector

2. The `dplyr` pipe:

- Groups the tokenized dataframe back into the original chunks using their unique identifiers
- Aggregates all individual word tokens into a single list for each chunk.
- Applies the `get_chunk_embedding()` function to each chunk's list of words.
- Expands the resulting 100-dimensional vector from a single list-column into 100 distinct columns (e.g., `embedding_dim_1`, `embedding_dim_2`, ... `embedding_dim_100`).

Question 1: Semantic Shift via Interruptions:

The first main question to address is:

Do interruptions shift the semantic meaning of an advocate's argument?

Add interruption labels

```
# Merge embedding vectors with metadata and label interruption status
df_meta_embeds <- df_unique %>% # before unnesting and tokenizing
  select(case_id, utt_id_first, utt_id_last,
         advocate_name, num_adv_utts_interrupted) %>%
  left_join(df_embeddings, by = c("case_id", "utt_id_first", "utt_id_last")) %>%
  mutate(interrupted = ifelse(num_adv_utts_interrupted > 0, "interrupted", "not_interrupted"))
```

To answer our first research question, we need to compare the semantics of interrupted vs. non-interrupted speech. For the comparison, it starts with the unique chunk metadata, selects the key identifiers and advocate information, and then merge in the 100-dimensional embedding vectors we just created (`df_embeddings`).

Finally, it creates the primary categorical variable for this analysis, `interrupted`. This new binary variable labels each chunk as either ‘interrupted’ or ‘not_interrupted’ based on the pre-computed `num_adv_utts_interrupted` column from the original dataset: if an utterance’s script presents any symbol for interruption, it will automatically be labeled as ‘interrupted’ (greater than 0).

In the transcript data, such interruptions are typically marked by double dashes (“—”) to indicate an abrupt cutoff or ellipses (“. . .”) to represent trailing speech.

Group chunks by advocate & interruption status

```
# Compute average embedding vector for each advocate by interruption status
advocate_embeddings <- df_meta_embeds %>%
  filter(!is.na(embedding_dim_V2)) %>% # check if embedding exists
  group_by(advocate_name, interrupted) %>%
  summarise(across(starts_with("embedding_dim_V"), mean), .groups = "drop")

# Reshape embeddings to wide format, separating interrupted vs. not_interrupted vectors
advocate_embeddings_wide <- advocate_embeddings %>%
  pivot_wider(
    names_from = interrupted,
    values_from = starts_with("embedding_dim"),
    names_prefix = "int_"
  )
```

To prepare for the semantic comparison, we aggregate the chunk-level embeddings to the advocate-level. It groups all valid chunks by `advocate_name` and `interrupted` status; then calculates the *mean 100-dimensional vector* for each advocate’s “interrupted” speech and “not_interrupted” speech, placing the mean ‘interrupted’ vector and the mean ‘not_interrupted’ vector side-by-side in separate columns.

Define cosine similarity and visualization

```
# Define helper function to compute cosine similarity between two embedding vectors
cosine_similarity <- function(a, b) {
  sum(a * b) / (sqrt(sum(a * a)) * sqrt(sum(b * b)))
}
```

```

# Get all column names
all_cols <- colnames(advocate_embeddings_wide)

# Filter the ones ending with _int_interrupted and _int_not_interrupted
int_1_cols <- grep("_int_interrupted$", all_cols, value = TRUE)
int_0_cols <- grep("_int_not_interrupted$", all_cols, value = TRUE)

# Sort to ensure same order
int_1_cols <- sort(int_1_cols)
int_0_cols <- sort(int_0_cols)

# Compute cosine similarity between each advocate's interrupted and non-interrupted embeddings
advocate_embeddings_wide <- advocate_embeddings_wide %>%
  rowwise() %>%
  mutate(
    cosine_sim = sim2(
      x = matrix(c_across(all_of(int_0_cols)), nrow = 1),
      y = matrix(c_across(all_of(int_1_cols)), nrow = 1),
      method = "cosine"
    )[1, 1]
  ) %>%
  ungroup()

# Summarize overall semantic similarity between interrupted and non-interrupted speech
mean(advocate_embeddings_wide$cosine_sim, na.rm = TRUE)

## [1] 0.9970004

summary(advocate_embeddings_wide$cosine_sim)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## 0.8899 0.9965 0.9977 0.9970 0.9986 0.9999      65

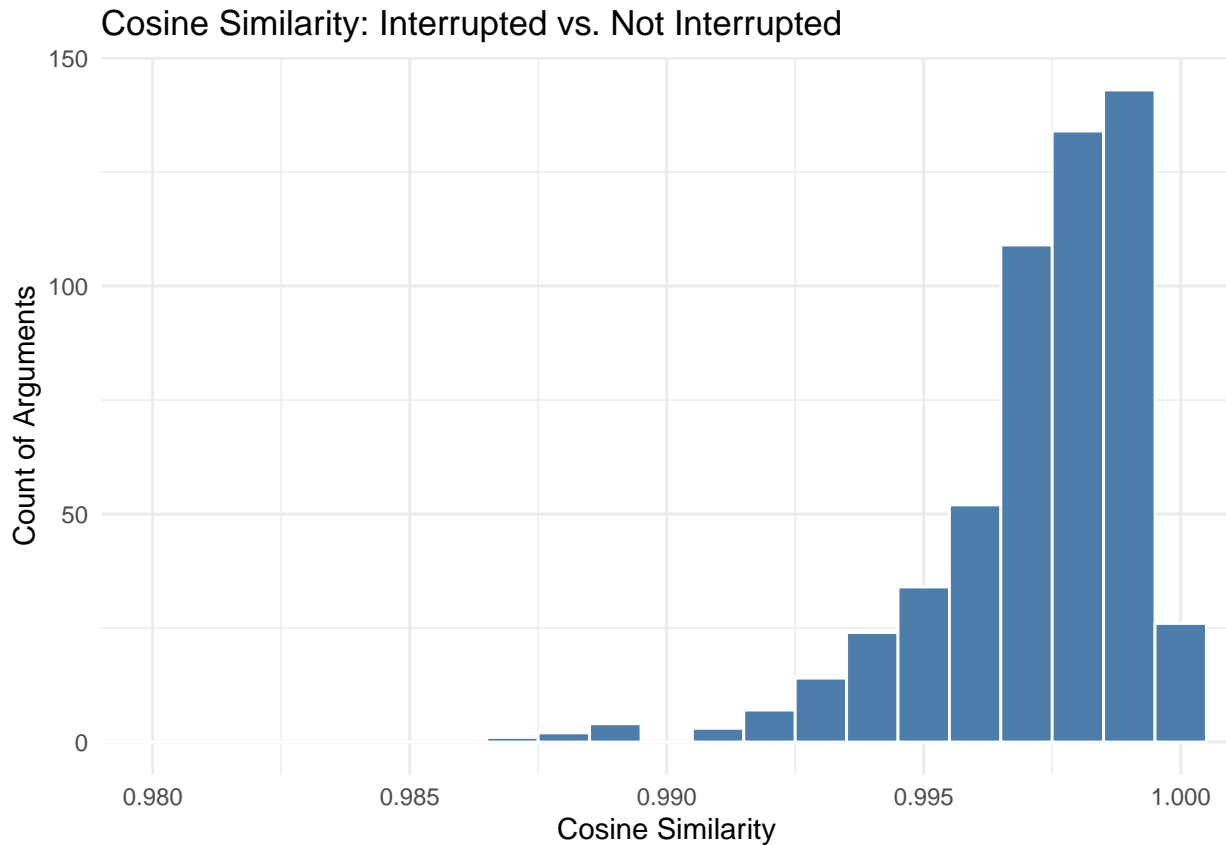
```

The mean cosine similarity is exceptionally high at 0.997. The distribution is extremely narrow and clustered near 1.0, with 50% of all advocates (the interquartile range) falling between 0.9965 and 0.9986. This indicates that semantic content is almost perfectly preserved, and interruptions do not significantly alter the core meaning of an advocate's speech.

```

# Plot distribution of cosine similarities between interrupted and non-interrupted arguments
ggplot(advocate_embeddings_wide, aes(x = cosine_sim)) +
  geom_histogram(binwidth = 0.001, fill = "#4D7EAB", color = "white") +
  coord_cartesian(xlim = c(0.98, 1)) + # zoom into dense range
  theme_minimal() +
  labs(
    title = "Cosine Similarity: Interrupted vs. Not Interrupted",
    x = "Cosine Similarity",
    y = "Count of Arguments"
  )

```



This histogram visually confirms the findings from the summary statistics. By zooming in on the 0.98 to 1.0 range, the plot clearly shows the distribution is heavily skewed to the right, with the vast majority of advocates having a cosine similarity score near-perfectly close to 1.0. This provides strong visual evidence for the conclusion that semantic content is preserved, as there is very little shift in meaning between interrupted and non-interrupted speech.

```
# Merge metadata and embedding data with original text for interpretation
df_with_text <- df_meta_embeds %>%
  left_join(df_unique %>%
    select(case_id, utt_id_first, chunk_text),
    by = c("case_id", "utt_id_first"))
```

```
# Find the lowest cosine similarity row
lowest <- advocate_embeddings_wide %>%
  filter(!is.na(cosine_sim)) %>%
  arrange(cosine_sim) %>%
  slice(1)
```

```
# Pull both types of chunks for that advocate
df_with_text %>%
  filter(advocate_name == lowest$advocate_name) %>%
  select(interrupted, chunk_text)
```

```
## # A tibble: 4 x 2
##   interrupted    chunk_text
##   <chr>          <chr>
## 1 interrupted    "Your Honor, in -- in this case, in Valley, there is some cir~
## 2 interrupted    "What -- what you're doing -- what you're doing, Your Honor, ~"
```

```
## 3 interrupted      "Absolutely, Your Honor.\nIt fits under Penn Central.\nAnd in~
## 4 not_interrupted "Mr. Chief Justice, may I answer the question? He wanted one ~
lowest$cosine_sim

## [1] 0.8899362
```

Conclusions and summary

1. Semantic content is largely preserved

The cosine similarity between interrupted and uninterrupted chunks is extremely high (mean approximately 0.997), with even the lowest similarity still reflecting substantial semantic overlap. This indicates that, across 12,663 speech segments, interruptions do **not** meaningfully alter what advocates are arguing. After being interrupted, advocates resume their statements with nearly identical semantic content as before, suggesting that interruptions rarely shift the underlying legal reasoning or substantive claims being presented.

2. Interruptions affect delivery rather than argument substance

Although interruptions do not modify semantic meaning, they likely disrupt the presentation, pacing, and rhetorical flow of an advocate's argument. This aligns with findings in judicial communication research that interruptions function more as expressions of interactional dominance or conversational control than as mechanisms that force conceptual reframing. Thus, the impact of interruptions appears to be **procedural rather than substantive**: shaping *how* arguments unfold rather than *what* is being argued.

Question 2: Gender effects on interruptions

Main Question 2:

Are interruptions directed at female advocates characterized by more negative sentiment than those directed at male counterparts?

To answer the question, we apply the NRC sentiment lexicon to assign emotion categories to each justice interruption. We then quantify the emotional content of interruptions directed at male versus female advocates, focusing on negative emotions (e.g., anger, disgust). Statistical comparisons test whether interruptions toward female advocates tend to be more emotionally negative, supporting our hypothesis of gendered treatment.

Preparation

Filter interrupted chunks only

These are where the **advocate was interrupted**, and we're interested in what the **justice says**.

```
# Filter only chunks where the advocate was interrupted
interrupted_chunks <- df_cleaned %>%
  filter(num_adv_utts_interrupted > 0)
```

Tokenize the chunk text (already cleaned)

```
# Tokenize interrupted chunks for sentiment analysis
tokens_sentiment <- interrupted_chunks %>%
  select(case_id, utt_id_first, advocate_name, advocate_gender, chunk_text_clean) %>%
  unnest_tokens(word, chunk_text_clean)
```

Load the NRC Lexicon

We use the NRC sentiment lexicon because it is specifically designed for the task of identifying *emotional tone*, which is the focus of our second research question. Unlike simpler lexicons that only provide a positive/negative polarity, the NRC lexicon (provided via `tidytext`) categorizes words into eight basic emotions (like ‘anger’, ‘disgust’, ‘fear’, ‘sadness’) as well as positive/negative. This allows us to move beyond a simple polarity score and directly quantify the “negative sentiment” (e.g., the proportion of anger- or disgust-related words) in interruptions, which is necessary to test our hypothesis about gendered treatment.

```
nrc <- get_sentiments("nrc")
```

Join Tokens with NRC

```
# Match tokens with NRC sentiment categories
tokens_labeled <- tokens_sentiment %>%
  inner_join(nrc, by = "word")
```

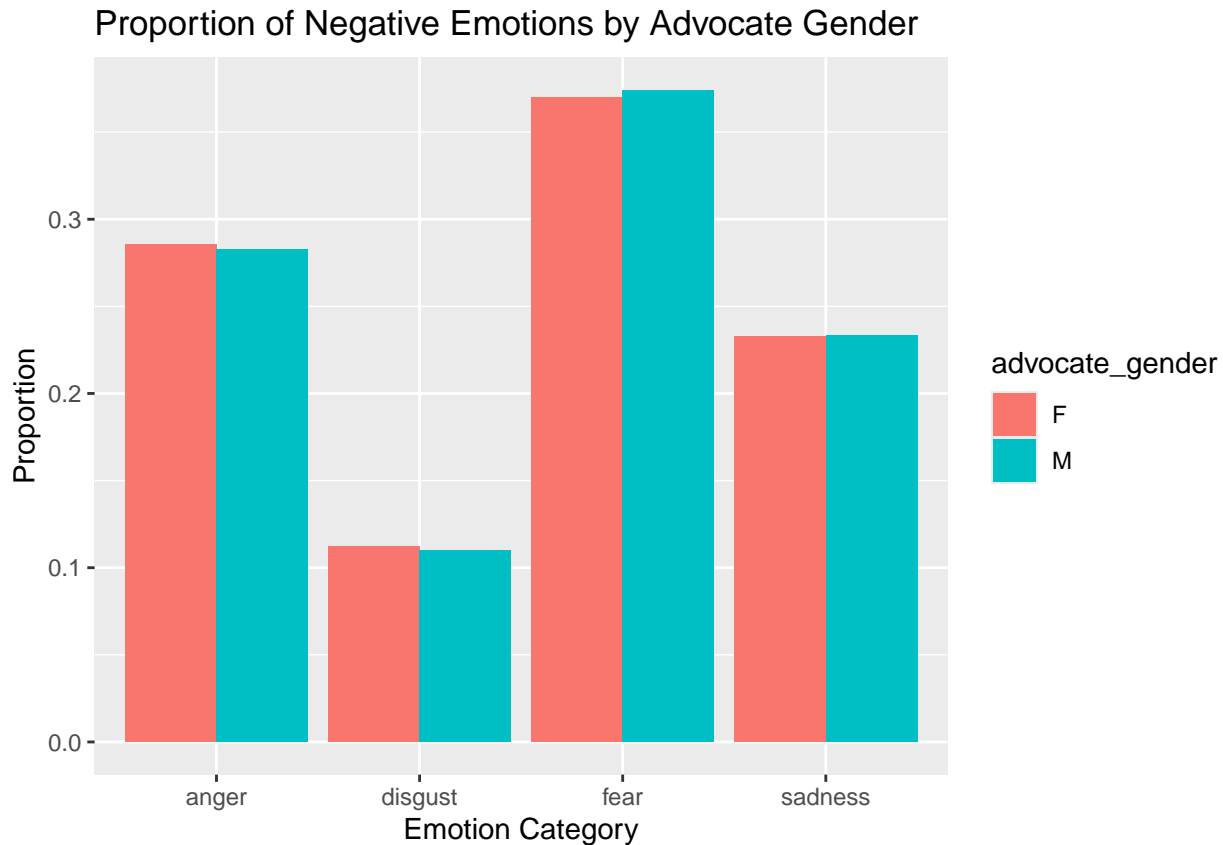
Visualization

```
# generate negative emotions only
negative_emotions <- c("anger", "fear", "disgust", "sadness")

# Calculate proportion of negative emotion words by advocate gender
emotion_summary_negative <- tokens_labeled %>%
  filter(sentiment %in% negative_emotions) %>%
  count(advocate_gender, sentiment) %>%
  group_by(advocate_gender) %>%
  mutate(prop = n / sum(n)) %>%
  ungroup()
```

Negative Emotion Composition by Gender

```
# Visualize proportion of negative emotions by advocate gender
ggplot(emotion_summary_negative, aes(x = sentiment, y = prop, fill = advocate_gender)) +
  geom_col(position = "dodge") +
  labs(title = "Proportion of Negative Emotions by Advocate Gender",
       y = "Proportion", x = "Emotion Category")
```



The distribution of negative emotion categories—anger, disgust, fear, and sadness—appears broadly similar for interruptions directed at male and female advocates, with nearly overlapping proportions across all four categories. While female advocates show slightly higher levels of anger and fear in these interruptions, the differences are small and visually subtle. Overall, the plot suggests that *the emotional composition of negative language used in interruptions is largely comparable across genders*, and that any gender differences arise from small shifts in overall negativity rather than from distinct patterns in specific negative emotions.

Welch Two Sample t-test

```
# Quantify emotion scores per chunk
emotion_scores <- tokens_labeled %>%
  filter(sentiment %in% c("positive", "negative", "anger", "disgust", "fear", "sadness")) %>%
  group_by(case_id, utt_id_first, advocate_name, advocate_gender, sentiment) %>%
  summarise(word_count = n(), .groups = "drop") %>%
  pivot_wider(
    names_from = sentiment,
    values_from = word_count,
    values_fill = 0
  )

# Normalize emotion scores
emotion_scores_normalized <- emotion_scores %>%
  mutate(
    total_emotion_words = positive + negative + anger + disgust + fear + sadness,
    neg_ratio = (negative + anger + disgust + fear + sadness) / total_emotion_words,
    pos_ratio = positive / total_emotion_words
  )
```

```
# Summarize average negative and positive sentiment ratios by advocate gender
emotion_gender_summary <- emotion_scores_normalized %>%
  group_by(advocate_gender) %>%
  summarise(
    avg_neg_ratio = mean(neg_ratio, na.rm = TRUE),
    avg_pos_ratio = mean(pos_ratio, na.rm = TRUE),
    n = n()
  )

emotion_gender_summary
```

```
## # A tibble: 2 x 4
##   advocate_gender avg_neg_ratio avg_pos_ratio     n
##   <chr>           <dbl>         <dbl> <int>
## 1 F               0.661           0.339  1362
## 2 M               0.647           0.353  8935
```

This summary table provides the descriptive statistics for our main variable of interest, the negative sentiment ratio. It shows that, on average, interruptions directed at female advocates (F) have a slightly higher mean negative sentiment ratio (0.661) than those directed at male advocates (M) (0.647).

```
t.test(neg_ratio ~ advocate_gender, data = emotion_scores_normalized)
```

```
##
## Welch Two Sample t-test
##
## data: neg_ratio by advocate_gender
## t = 2.6623, df = 1729.7, p-value = 0.007834
## alternative hypothesis: true difference in means between group F and group M is not equal to 0
## 95 percent confidence interval:
##  0.003876661 0.025572405
## sample estimates:
## mean in group F mean in group M
##      0.6614017      0.6466771
```

The Welch Two Sample t-test confirms that the small difference observed in the summary table is statistically significant. With a **p-value of 0.007834**, which is well below the standard 0.05 threshold, we can reject the null hypothesis. This test provides statistical evidence for our second research question, supporting the conclusion that interruptions directed at female advocates are, on average, characterized by a higher negative sentiment ratio than those directed at male advocates. The 95% confidence interval [0.0039, 0.0256] further confirms this, as it does not contain zero.

Robustness Check

```
# Combine sentiment scores with advocate and case-level covariates for modeling
df_model <- emotion_scores_normalized %>%
  left_join(
    df_cleaned %>% select(case_id, utt_id_first, advocate_gender, adv_experience_int, case_year, female)
    by = c("case_id", "utt_id_first")
  )

# Clean merged data and run OLS regression predicting negative sentiment ratio
df_model <- df_model %>%
  mutate(advocate_gender = coalesce(advocate_gender.x, advocate_gender.y)) %>%
  select(-advocate_gender.x, -advocate_gender.y)
```

```
model <- lm(neg_ratio ~ advocate_gender + adv_experience_int + case_year + female_issue + advocate_ideo.
summary(model)
```

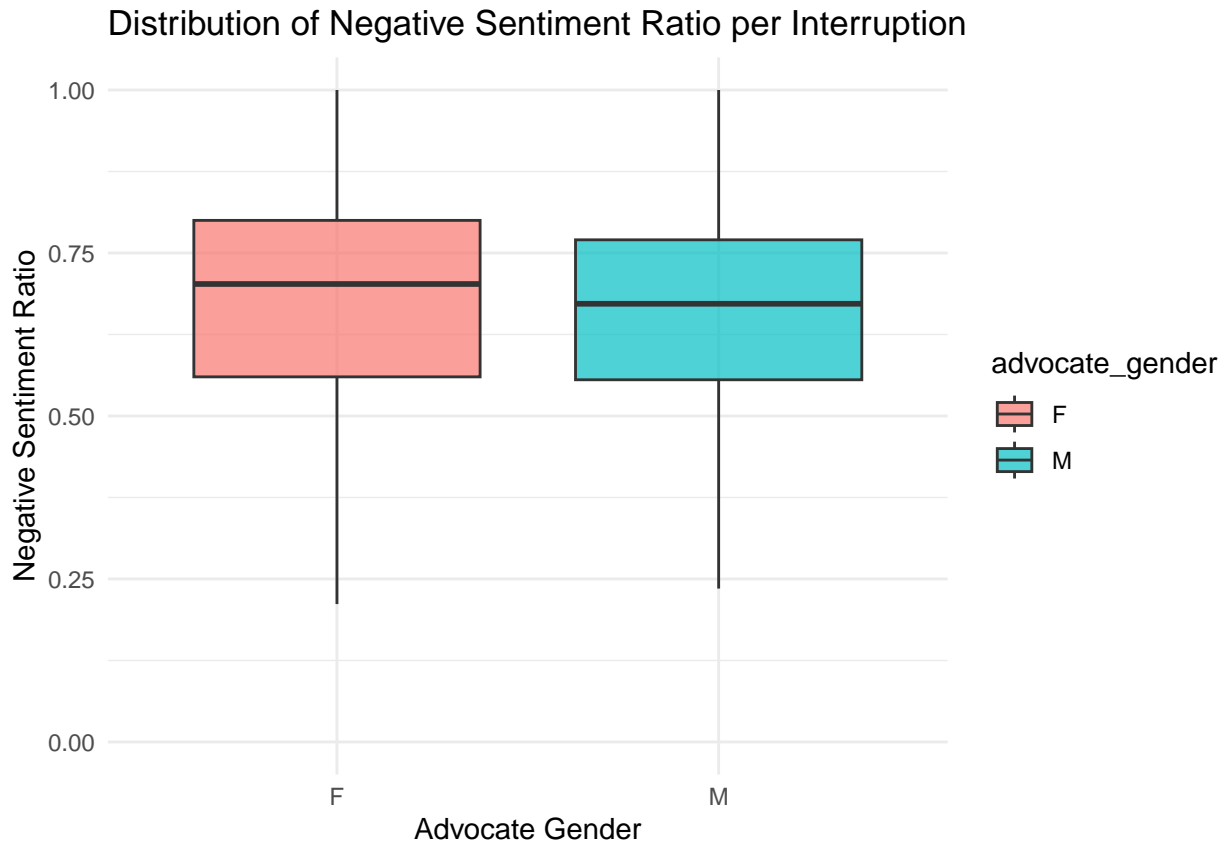
```
##
## Call:
## lm(formula = neg_ratio ~ advocate_gender + adv_experience_int +
##     case_year + female_issue + advocate_ideology + ideology_matches,
##     data = df_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67073 -0.09258  0.02782  0.12432  0.39225
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.4087998   1.2815897   -1.099   0.2717
## advocate_genderM    -0.0113790   0.0052349   -2.174   0.0298 *
## adv_experience_int    -0.0021587   0.0003996   -5.402 6.75e-08 ***
## case_year         0.0010275   0.0006364    1.615   0.1064
## female_issue              NA           NA         NA      NA
## advocate_ideologyliberal  0.0060803   0.0035424    1.716   0.0861 .
## ideology_matches      0.0017843   0.0035943    0.496   0.6196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1785 on 10291 degrees of freedom
## Multiple R-squared:  0.004113,    Adjusted R-squared:  0.00363
## F-statistic: 8.501 on 5 and 10291 DF,  p-value: 4.831e-08
```

The key variable, `advocate_genderM`, has a p-value of 0.0298, which is statistically significant. This indicates that even after controlling for advocate experience, case year, and ideology, the gender of the advocate remains a significant predictor of the negative sentiment ratio, supporting the project's second hypothesis that female advocates face more negative interruptions.

Distribution of Negativity Ratio by Gender

```
# Visualize distribution of negative sentiment ratios across advocate genders
df_with_ratios <- emotion_scores_normalized

ggplot(df_with_ratios, aes(x = advocate_gender, y = neg_ratio, fill = advocate_gender)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  labs(
    title = "Distribution of Negative Sentiment Ratio per Interruption",
    x = "Advocate Gender",
    y = "Negative Sentiment Ratio"
  ) +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal()
```



This boxplot provides a clear visual confirmation of the t-test’s findings. It illustrates the distribution of the negative sentiment ratio (`neg_ratio`) for interruptions by advocate gender. We can clearly see that the median (the center line) and the entire interquartile range (the box) for female advocates (F) are shifted slightly higher than for male advocates (M). This plot visually supports the statistically significant conclusion that interruptions directed at women have a higher average negative sentiment ratio.

Conclusion

Across both visualizations and statistical tests, interruptions directed at female advocates show a slightly higher level of negative sentiment than those directed at male advocates. The emotion-category bar plot indicates that the composition of anger, disgust, fear, and sadness is broadly similar across genders, but the boxplot shows that the overall negative sentiment ratio is shifted modestly higher for female advocates.

A Welch two-sample *t*-test confirms that this difference is statistically significant ($p = 0.0078$), and an OLS regression that controls for advocate experience, case year, and ideological factors yields the same conclusion ($p = 0.0298$). Together, these results indicate a small but consistent pattern: **interruptions toward female advocates contain a slightly higher proportion of negative emotional language than interruptions toward male advocates.**

Exploratory: Topic Modeling (LDA)

Our t-test and OLS regression found that interruptions directed at female advocates have a significantly higher negative sentiment ratio. However, a skeptic could argue this isn’t a true gender effect, but rather a *topic* effect: for instance, “What if women just happen to argue cases about different topics that naturally use more negative words?”

We are running this LDA to identify these underlying topics. The goal is to then add these topics as control variables in a final regression model, to see if the `advocate_gender` variable remains statistically significant.

If it does, it strengthens our conclusion that the gender disparity in sentiment is a genuine finding, not just an artifact of different case subjects.

```
# Create document-term matrix (DTM) for topic modeling
lda_input <- df_cleaned %>%
  select(utt_id_first, chunk_text_clean) %>%
  unnest_tokens(word, chunk_text_clean) %>%
  anti_join(stop_words, by = "word") %>%
  count(utt_id_first, word) %>%
  cast_dtm(utt_id_first, word, n)

# Fit Latent Dirichlet Allocation (LDA) model with 6 topics
lda_model <- LDA(lda_input, k = 6, control = list(seed = 42))

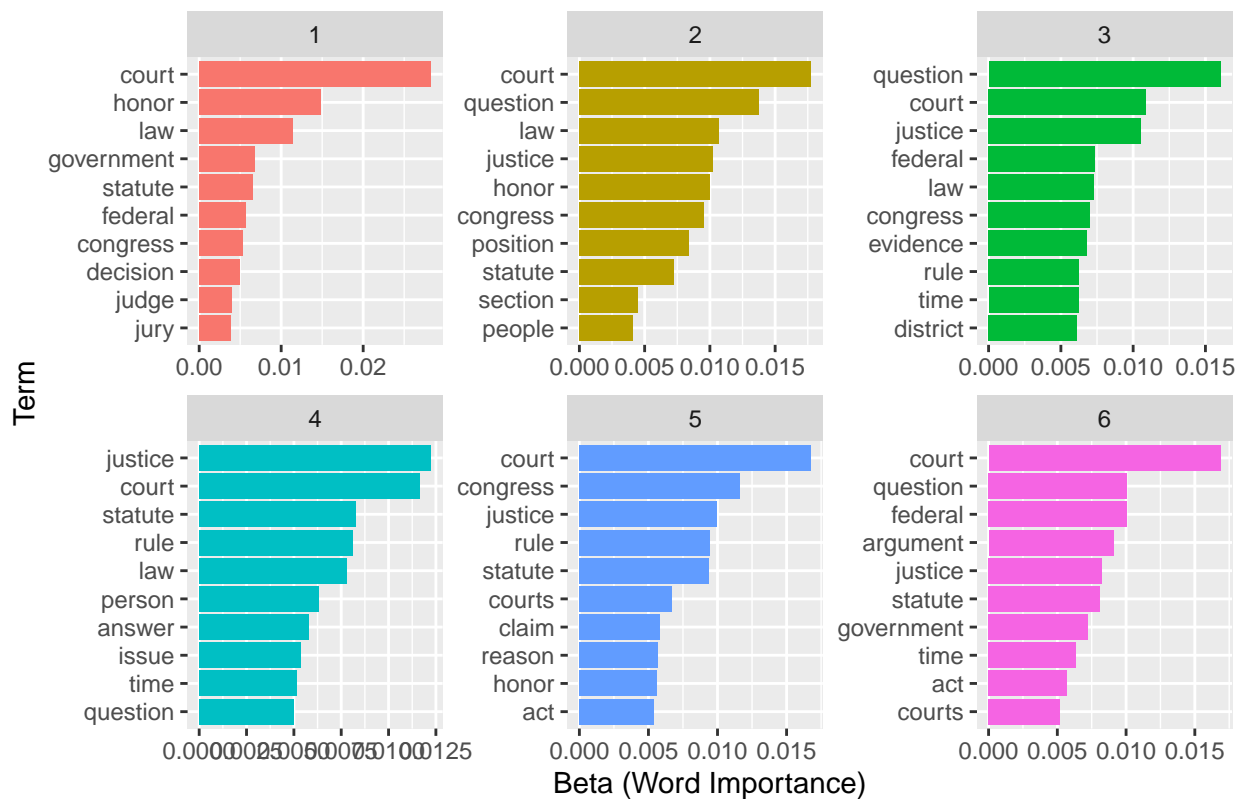
chunk_topics <- tidy(lda_model, matrix = "gamma") # per-doc topic weights

# Merge topic probabilities back to original metadata
topic_df <- chunk_topics %>%
  pivot_wider(names_from = topic, values_from = gamma, names_prefix = "topic_") %>%
  left_join(df_cleaned, by = c("document" = "utt_id_first"))

# Extract topic probabilities for each speech chunk and merge with metadata
top_terms <- tidy(lda_model, matrix = "beta") %>% # per-word topic probabilities
  group_by(topic) %>%
  top_n(10, beta) %>% # top 10 words
  ungroup() %>%
  arrange(topic, -beta)

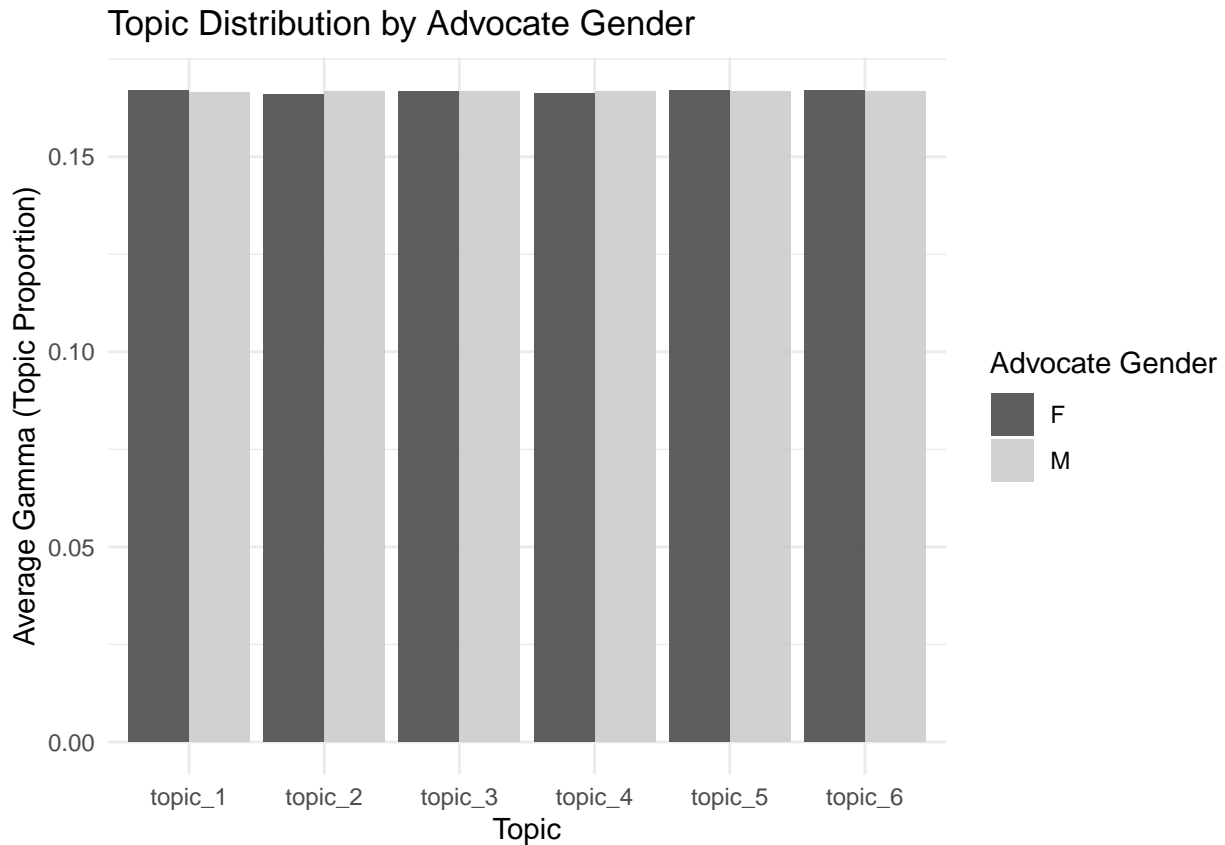
ggplot(top_terms, aes(x = reorder_within(term, beta, topic), y = beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_x_reordered() +
  coord_flip() +
  labs(x = "Term", y = "Beta (Word Importance)", title = "Top 10 Words per Topic")
```

Top 10 Words per Topic



```
# Compare average topic proportions by advocate gender and visualize results
gender_topics <- topic_df %>%
  filter(!is.na(advocate_gender)) %>% # Remove rows with NA gender
  group_by(advocate_gender) %>%
  summarize(across(starts_with("topic_"), \(x) mean(x, na.rm = TRUE))) %>%
  pivot_longer(
    cols = starts_with("topic_"),
    names_to = "topic",
    values_to = "avg_gamma"
  )

# Plot topic distribution by gender
ggplot(gender_topics, aes(x = topic, y = avg_gamma, fill = advocate_gender)) +
  geom_col(position = "dodge", alpha = 0.9) +
  scale_fill_manual(values = c("F" = "gray30", "M" = "gray80")) +
  labs(
    x = "Topic",
    y = "Average Gamma (Topic Proportion)",
    title = "Topic Distribution by Advocate Gender",
    fill = "Advocate Gender"
  ) +
  theme_minimal()
```



The LDA robustness check proved inconclusive and did not add analytical value to the sentiment analysis. The model's failure stems from several issues:

First, the model produced **highly overlapping topics**. This was driven by a **dominated vocabulary**, where all 6 topics were saturated with the same high-frequency, generic "court talk" words (e.g., "court," "justice," "question"). As a result, there was **no clear thematic separation** between them; they failed to capture distinct conceptual themes and instead just represented minor variations of the same legal discourse. This suggests the formal, narrow, and structured nature of Supreme Court arguments **violated the assumptions** of the LDA model.

This failure was confirmed computationally: when the topic proportions were added to the OLS regression, the model produced singularities (e.g., **NA** coefficients), confirming the topics were not distinct and could not be used as a valid control.