Topic Modeling

Ruijian Maggie Lin

2024-11-01

Read & Clean Data

```
movies <- read.csv("movie_plots.csv")

# Unnesting tokens using tidytext
plots_by_words <- movies %>% unnest_tokens(word, Plot) %>%
  anti_join(stop_words) %>%
  count(Movie.Name, word, sort = TRUE)
```

```
Joining with `by = join_by(word)`
```

```
# Removing common first names using the 'lexicon package'
data("freq_first_names")
first_names <- tolower(freq_first_names$Name)
plot_word_counts <- plots_by_words %>% filter(!(word %in% first_names))
```

Determine optimal number of topics (k) with ldatuning & Scree Plot

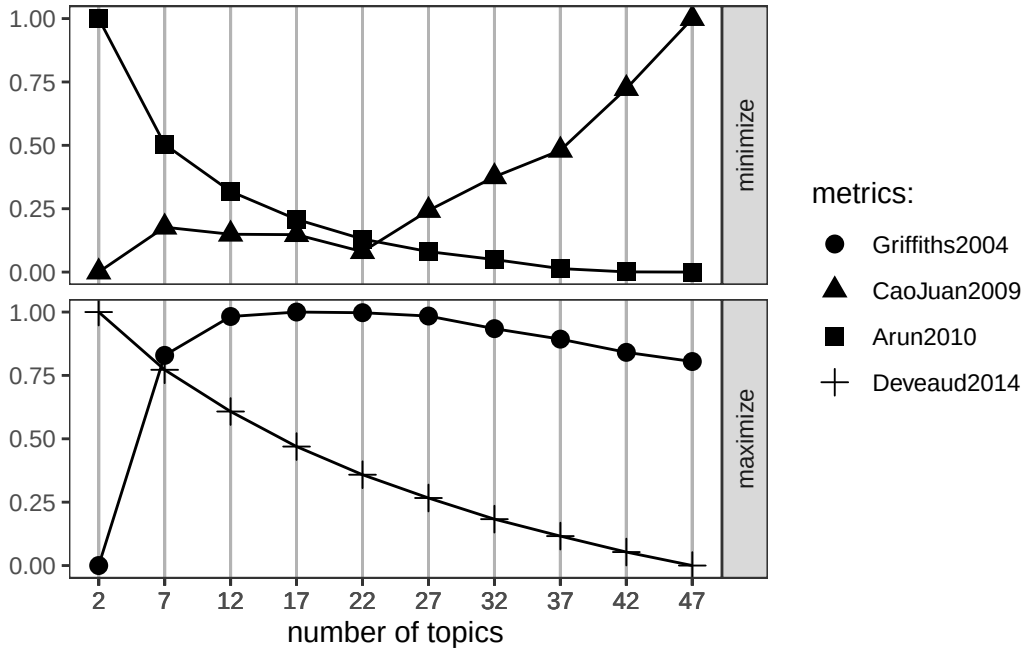
```
# Casting our word counts to a document term matrix
plots_dtm <- plot_word_counts %>% cast_dtm(Movie.Name, word, n)

result <- FindTopicsNumber(
  plots_dtm,
  topics = seq(2, 50, by = 5),    # Test from 2 to 50 topics in increments of 5
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 1234),
```

```
  mc.cores = 2L,
  verbose = TRUE
)
```

```
fit models... done.
calculate metrics:
  Griffiths2004... done.
  CaoJuan2009... done.
  Arun2010... done.
  Deveaud2014... done.
```

```
FindTopicsNumber_plot(result)
```

```
Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
of ggplot2 3.3.4.
i The deprecated feature was likely used in the ldatuning package.
  Please report the issue at <https://github.com/nikita-moor/ldatuning/issues>.
```




```
# Use the optimal k value from the plot to proceed with topic modeling.

plot_word_counts_tfidf <- plot_word_counts %>%
```

```
  bind_tf_idf(word, Movie.Name, n) %>%
  filter(tf_idf > 0) %>%
  cast_dtm(Movie.Name, word, tf_idf)
```

INTERPRETATION

Griffiths2004

- This metric evaluates model likelihood, and higher values indicate a better fit. It is maximized, meaning that looking for the peak in this line.
- In the plot, it (circles) is reach a high point around k = 7.

CaoJuan2009

- This metric calculates topic coherence, specifically looking at the pairwise similarity of topics. Lower values indicate less overlap (better topic separation). It is minimized, meaning that looking for the lowest point in the line.
- In the plot, it (triangles) is lowest around k = 7, suggesting that topics are most distinct at this value.

Arun2010

- This metric compares the distributions of document-topic and topic-word matrices, with lower values indicating a better fit. It measures the divergence between these distributions. It is minimized, meaning that looking for the lowest point in the line.
- In the plot, it (squares) is fairly low and stable around k = 7 but does not have a clear optimal point beyond that.

Deveaud2014

- This metric evaluates topic coherence based on word similarity within each topic. Higher values indicate better coherence. It is maximized, meaning that looking for the peak in this line.
- In the plot, it (crosses) peaks around k = 7 to k = 12, suggesting that topic coherence is strong in this range.

Scree Plot helps determine the optimal number of topics by showing coherence scores.

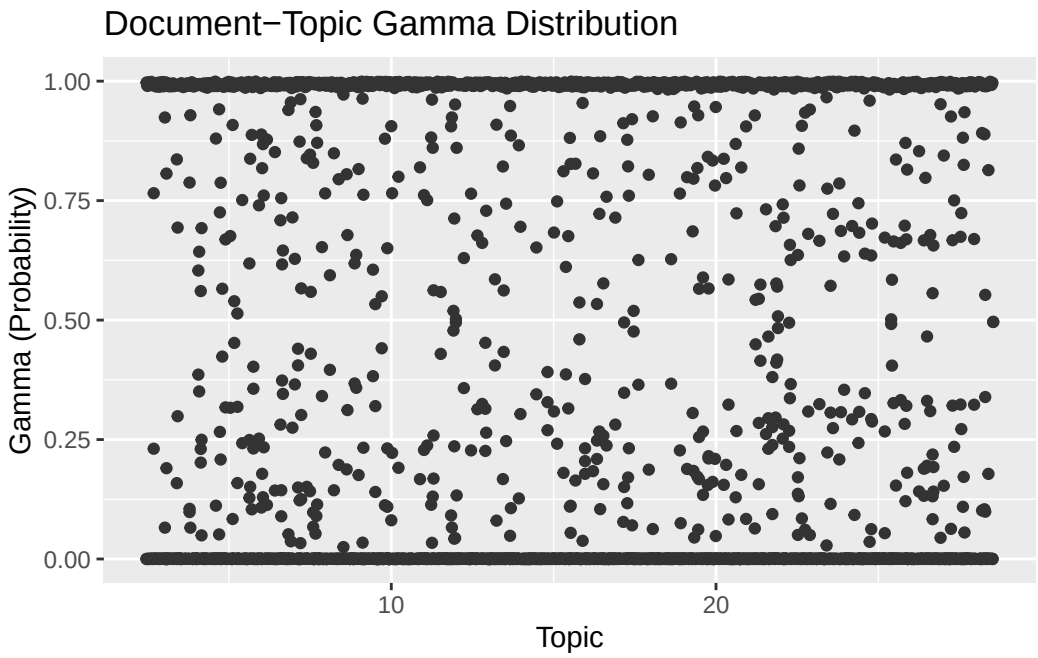
Most metrics suggest that a topic count of around k = 7 might be optimal, as it balances topic coherence and separation. I can select this as the number of topics, or I could experiment with values in the range of 7 to 12 for slightly more nuanced topic distinctions.

Perform topic modeling with LDA

```
# LDA with 30 topics
plots_lda <- LDA(plots_dtm, k = 30, control = list(seed = 1106))
```

Gamma Plot

```
plots_gamma <- tidy(plots_lda, matrix = "gamma")

# Gamma Plot (Document-Topic Probabilities)
plots_gamma %>%
  ggplot(aes(topic, gamma, fill = factor(document))) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Document-Topic Gamma Distribution", x = "Topic", y = "Gamma (Probability)")
```



INTERPRETATION

High Gamma Values: The points near the top (gamma close to 1) suggest documents that are highly associated with a single topic.

Low Gamma Values: Points closer to the bottom (gamma close to 0) indicate weak associations with that topic.

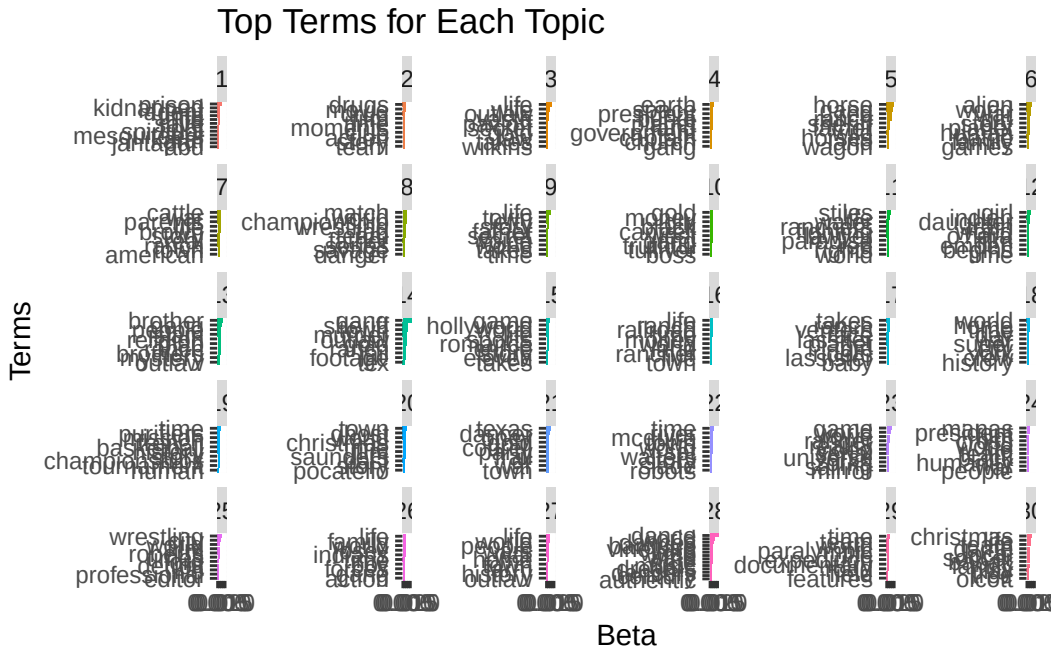
Mixed Gamma Values: The scatter of dots across different gamma values suggests that some documents may have mixed associations across topics, especially if there are points at intermediate values (e.g., around 0.5).

- The dense line of points at gamma = 1 shows that many documents have a strong association with at least one topic, which can indicate well-defined topics.
- The scatter below this line suggests that some documents are moderately or weakly associated with additional topics, which could imply some topic overlap or documents that don't fit neatly into a single category.

Gamma Plot shows how strongly each movie aligns with each topic, and it helps assess how distinctly documents are assigned to topics and whether there is overlap. Well-separated topics tend to have more gamma values close to 1 for individual documents, while more evenly spread values across topics indicate overlap or ambiguity in topic distinctions.

Beta Plot

```
plots_beta <- tidy(plots_lda, matrix = "beta")

top_terms <- plots_beta %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)

# Beta Plot (Word-Topic Probabilities)
top_terms %>%
  ggplot(aes(beta, reorder_within(term, beta, topic), fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y") +
  labs(title = "Top Terms for Each Topic", x = "Beta", y = "Terms") +
  scale_y_reordered()
```



INTERPRETATION

For each topic, the terms with the longest bars are the ones that are most descriptive or characteristic of that topic. For example, in Topic 1, "parents," "war," and "closer" may be the most descriptive terms.

Some terms appear in multiple topics but with varying beta values, suggesting that while a term might be relevant to multiple topics, its strength of association varies. For example, "game" might appear in several topics but be more strongly associated with a particular one.

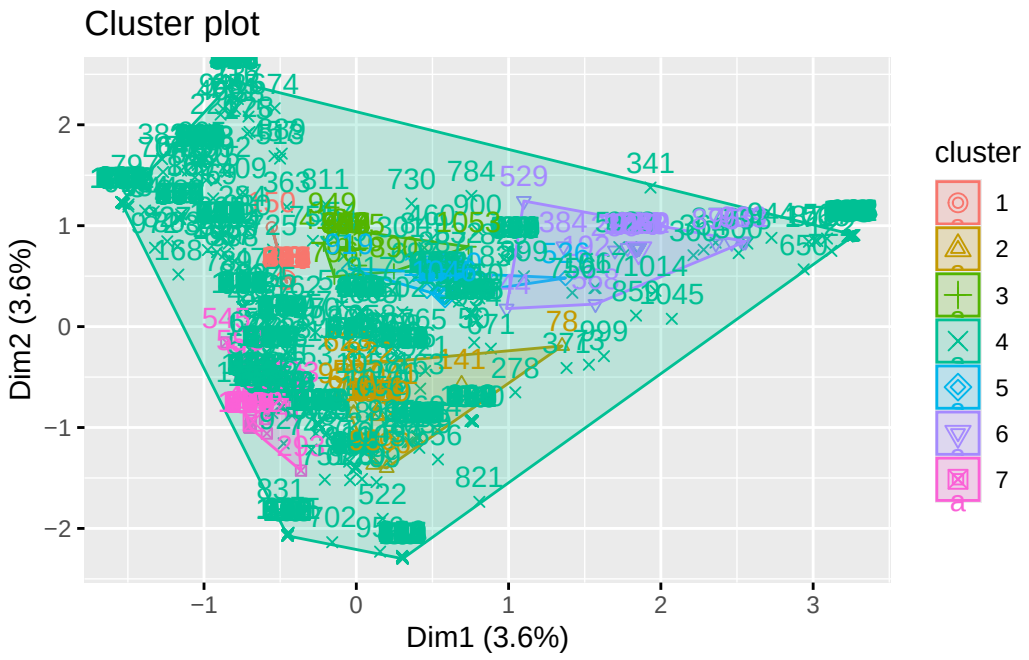
- Topics with distinct terms (less overlap in terms with other topics) suggest well-separated topics in the model.
- If multiple topics have overlapping terms, it may indicate that these topics are related or that the topic boundaries are less distinct.

Beta Plot highlights which words best represent each topic, giving insight into the model's interpretation of themes in the data. Using the information to label or categorize topics based on their top terms, aid in summarizing or communicating the topics extracted from the dataset.

Clustering with K-means and Cluster Plot

Clustering 7 genres (7 clusters)

```
# Pivoting the plots_gamma table wider to cluster by gammas for each topic
plots_gamma_wider <- plots_gamma %>% pivot_wider(
  names_from = topic,
  values_from = gamma
) %>%
  drop_na()

set.seed(1106)
cluster <- kmeans(plots_gamma_wider %>% select(-document), 7)
fviz_cluster(cluster, data = plots_gamma_wider %>% select(-document))
```




```
# Look at the genres in each cluster; read data contains genres information
english_movies_with_genres <- read.csv("movie_plots_with_genres.csv")
clusters <- cluster[["cluster"]]
plots_gamma_wider$cluster <- clusters

# Create an empty list to store genre counts for each cluster
cluster_genre_counts <- list()

set.seed(1106)
# Loop over each cluster and analyze genre composition
for (i in 1:7) {
```

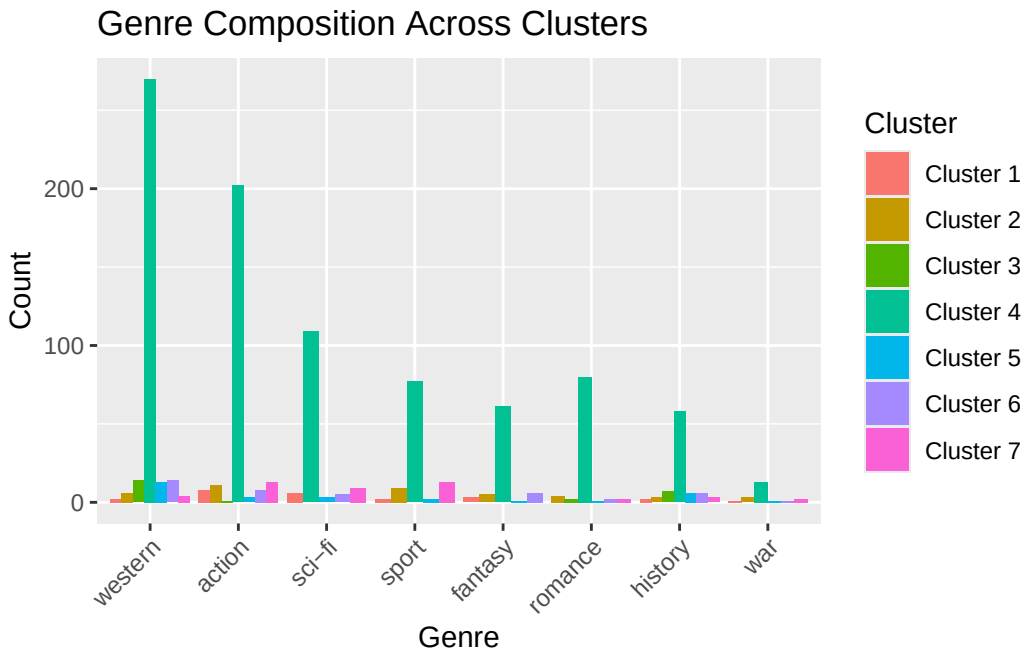
```r
  # Filter movies in the current cluster
  plots_cluster <- plots_gamma_wider %>% filter(cluster == i)

  # Extract movie names in this cluster
  cluster_names <- plots_cluster$document

  # Filter the original dataset for movies in this cluster and count genres
  cluster_data <- english_movies_with_genres %>%
    filter(Movie.Name %in% cluster_names)

  # Summarize the count of each genre in the current cluster
  cluster_counts <- cluster_data %>%
    group_by(Genre) %>%
    summarize(n = n()) %>%
    arrange(desc(n))

  # Store the genre counts in the list with the cluster number as the name
  cluster_genre_counts[[paste("Cluster", i)]] <- cluster_counts

  # Print the genre counts for the current cluster
  print(paste("Genre composition for Cluster", i))
  print(cluster_counts)
}
```

```
[1] "Genre composition for Cluster 1"
# A tibble: 7 x 2
  Genre       n
  <chr>   <int>
1 action      8
2 sci-fi      6
3 fantasy     3
4 history     2
5 sport       2
6 western     2
7 war         1
[1] "Genre composition for Cluster 2"
# A tibble: 7 x 2
  Genre       n
  <chr>   <int>
1 action     11
2 sport       9
3 western     6
```

```
4 fantasy     5
5 romance     4
6 history     3
7 war         3
[1] "Genre composition for Cluster 3"
# A tibble: 4 x 2
  Genre       n
  <chr>   <int>
1 western    14
2 history     7
3 romance     2
4 action      1
[1] "Genre composition for Cluster 4"
# A tibble: 8 x 2
  Genre       n
  <chr>   <int>
1 western   270
2 action    202
3 sci-fi    109
4 romance    80
5 sport      77
6 fantasy    61
7 history    58
8 war        13
[1] "Genre composition for Cluster 5"
# A tibble: 8 x 2
  Genre       n
  <chr>   <int>
1 western    13
2 history     6
3 action      3
4 sci-fi      3
5 sport       2
6 fantasy     1
7 romance     1
8 war         1
[1] "Genre composition for Cluster 6"
# A tibble: 7 x 2
  Genre       n
  <chr>   <int>
1 western    14
2 action      8
3 fantasy     6
```

```
4 history     6
5 sci-fi      5
6 romance     2
7 war         1
[1] "Genre composition for Cluster 7"
# A tibble: 7 x 2
  Genre       n
  <chr>   <int>
1 action     13
2 sport      13
3 sci-fi      9
4 western     4
5 history     3
6 romance     2
7 war         2
```

```r
# Combine all cluster genre counts into one data frame for visualization
combined_cluster_counts <- bind_rows(
  lapply(names(cluster_genre_counts), function(cluster_name) {
    cluster_genre_counts[[cluster_name]] %>%
      mutate(Cluster = cluster_name)
  })
)

# Plotting Genre Distribution Across Clusters
combined_cluster_counts %>%
  ggplot(aes(x = reorder(Genre, -n), y = n, fill = Cluster)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Genre Composition Across Clusters",
    x = "Genre",
    y = "Count"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



INTERPRETATION

Cluster Plot:

- The plot groups data points into seven clusters, each represented by a different shape and color. Each cluster might represent a distinct type of media or genre grouping, reflecting similarity among items in that group.
- There is some overlap between clusters, suggesting that some items have mixed or similar features across categories, while other items are more distinct and unique to their group.
- The axes (Dim1 and Dim2) explain only a small percentage of variance (3.6% each), which might mean that higher-dimensional features contribute more to the clustering, and these two dimensions provide a simplified view.

Genre Composition Across Clusters:

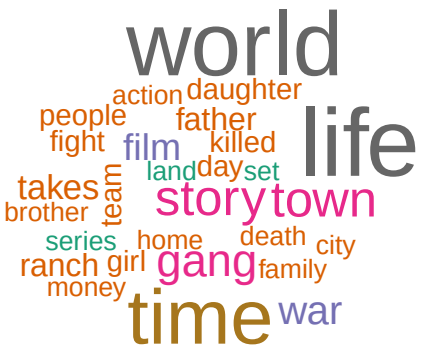
- The green cluster (Cluster 4) appears to dominate in most genres, especially in "western," "action," "sci-fi," "fantasy," "romance," and "history." This may indicate that Cluster 4 encompasses a broad range of genres or represents the majority of the data points.
- Other clusters have smaller genre counts and may represent more niche groupings or specific subgenres.
- Genres such as "western," "action," and "sci-fi" are the most frequently occurring across all clusters, possibly indicating that these genres have common features shared with various other genres.

Artsy Word Cloud

```
wordcloud_data <- plot_word_counts %>%
  inner_join(plots_beta, by = c("word" = "term")) %>%
  group_by(word) %>%
  summarize(frequency = sum(beta)) %>%
  arrange(desc(frequency))
```

```
Warning in inner_join(., plots_beta, by = c(word = "term")): Detected an unexpected many-to-m
i Row 1 of `x` matches multiple rows in `y`.
i Row 961 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
wordcloud(words = wordcloud_data$word, freq = wordcloud_data$frequency,
          max.words = 100, colors = brewer.pal(8, "Dark2"), scale = c(3, 0.5))
```



INTERPRETATION

Larger words indicate a higher probability of appearing in the topics generated by the model.

- The largest words, such as “world,” “life,” “story,” “time,” “gang,” and “town,” are the most significant in the dataset. Their large size suggests they are frequently mentioned or have high importance across the topics identified by the model.
- These terms likely represent common themes or concepts in the data, indicating that the topics often revolve around ideas of life events, narratives, world settings, temporal aspects (“time”), and community or social groups (“gang,” “town”).

The combination of these words paints a picture of narrative content that might explore various human experiences and relationships, particularly those involving family, conflict, social groups (like gangs or teams), and survival or life challenges.

- The presence of words like “world” and “story” could imply that the data contains a variety of narratives or settings, possibly in diverse genres or locales, while “time” suggests an exploration of events over time or significant moments.