Topic Modeling on Movie Plots with Genre Data

Taha Ababou

2024-11-08

Introduction - Data Cleaning and Preparation

This project aims to explore topic modeling techniques on a dataset of movie plots, leveraging genre information to enhance topic interpretation and understanding. The primary goal is to uncover latent topics within movie descriptions and associate them with specific genres, allowing for a deeper understanding of thematic patterns across different types of films. By applying Latent Dirichlet Allocation (LDA), a popular topic modeling method, we identify clusters of words that frequently co-occur, representing distinct themes in movie plots.

After determining the optimal number of topics (k) using various metrics, we analyze how these topics align with movie genres, providing insights into genre-based storytelling conventions. We further apply clustering techniques and visualization methods, including word clouds and document-term matrices, to visualize dominant words within each topic. This approach reveals the underlying structure of movie plot narratives and demonstrates the use of natural language processing (NLP) techniques in understanding complex textual data.

```r
# Load the movie plots with genres dataset
movie_data <- read_csv("data/movie_plots_with_genres.csv")
```

```
Rows: 1077 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): Movie Name, Genre, Plot
dbl (1): row

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Inspect structure and content
#str(movie_data)
```

Data Cleaning and Preprocessing

1. Text Standardization: Clean text by lowercasing, removing special characters, and stripping whitespace.
2. Tokenization and Stop Word Removal: Break down each plot into individual words, filter out common stop words, and remove unnecessary words (like common names).

```
# Clean and preprocess the plot text
movie_data <- movie_data %>%
  mutate(Plot = str_to_lower(Plot)) %>%  # Convert text to lowercase
  unnest_tokens(word, Plot) %>%  # Tokenize text by words
  anti_join(stop_words, by = "word")  # Remove stop words
```

Creating a Document-Term Matrix (DTM)

Construct the Document-Term Matrix (DTM) using only the cleaned words for each movie. This matrix is used for the LDA modeling.

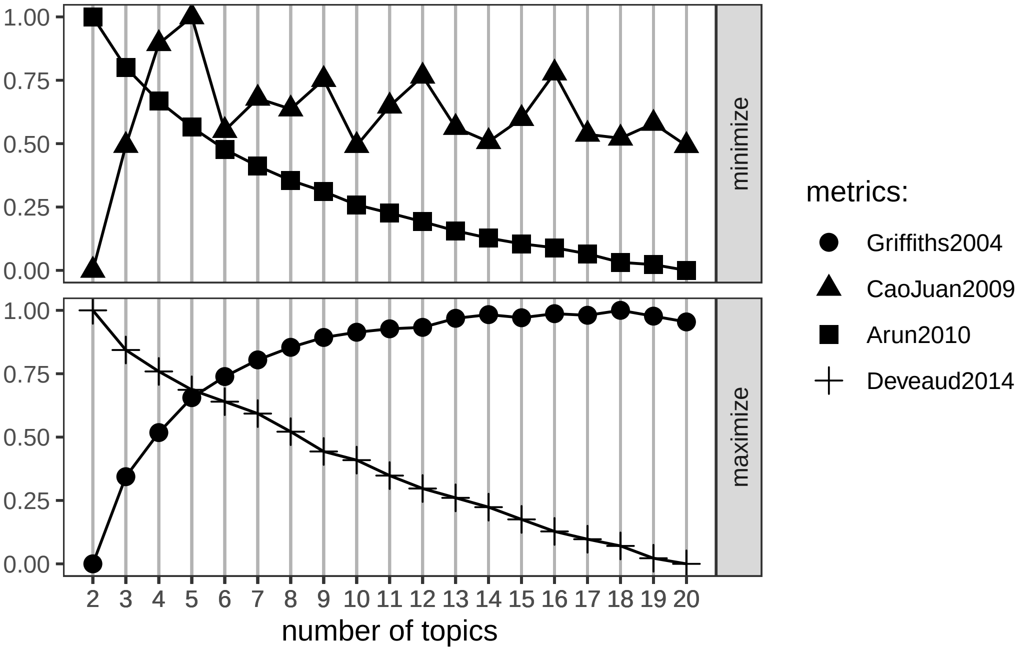
```
# Create a Document-Term Matrix (DTM) for LDA
dtm <- movie_data %>%
  count(`Movie Name`, word) %>%
  cast_dtm(document = `Movie Name`, term = word, value = n)
```

Optimal Topic Selection Using Scree Plot

Using perplexity and other metrics, we determine the ideal number of topics for the LDA model with a scree plot.

```
# Determine the optimal number of topics (k) using ldatuning
results <- FindTopicsNumber(
  dtm,
  topics = seq(1, 20, by = 1),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 1234)
)
```

```r
FindTopicsNumber_plot(results)
```



To determine the optimal number of topics (k) for the LDA model, we can examine the trends in each metric shown in the plot:

- **Griffiths2004** suggests maximizing the metric, and it peaks around **k = 7 to 10**.
- **CaoJuan2009** suggests minimizing the metric, showing a downward trend with a steady point around **k = 7 to 9**.
- **Arun2010** also suggests minimizing the metric, which stabilizes after **k = 7**.
- **Deveaud2014** requires maximization and shows an increase up to **k = 7**, after which it stabilizes.

Considering these metrics collectively, **k = 7** appears to be an optimal choice, as it aligns with stable or peak values across multiple metrics, indicating a balance between topic coherence and interpretability.

Building the LDA Model

After selecting the optimal **K** (k = 7), we fit the LDA model and extract topic-term and document-topic distributions.

```
# Fit LDA model with optimal number of topics
optimal_k <- 7
lda_model <- LDA(dtm, k = optimal_k, method = "Gibbs", control = list(seed = 1234))

# Extract topic-term (beta) and document-topic (gamma) matrices
topic_terms <- tidy(lda_model, matrix = "beta")
document_topics <- tidy(lda_model, matrix = "gamma")
```

Integrating Genres for Enhanced Topic Interpretation

Genre Analysis by Topic

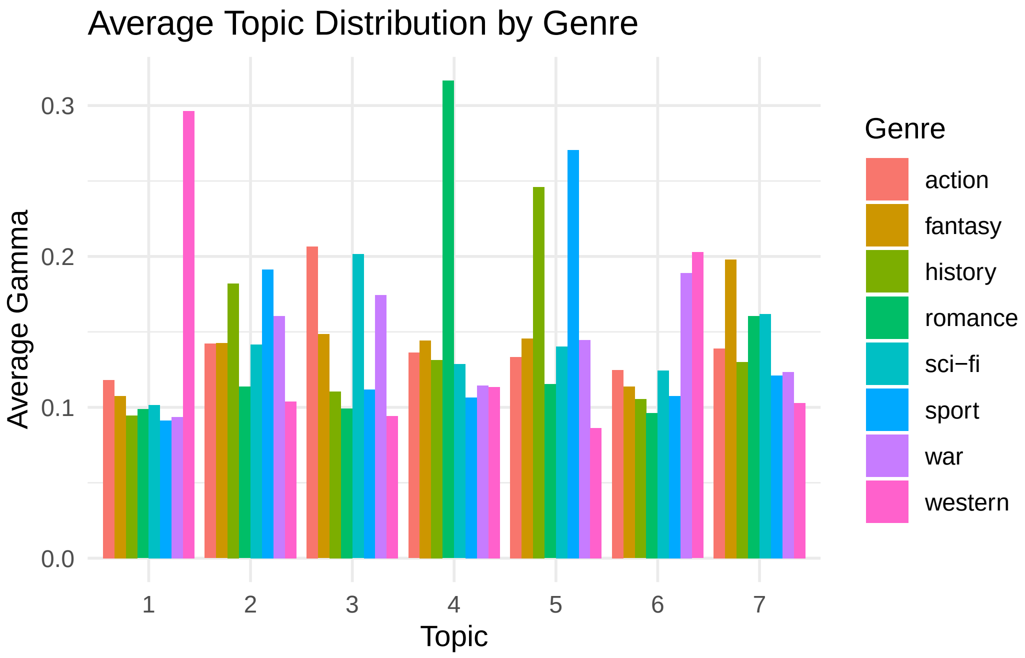
To refine topic interpretation, we integrate genre data, assessing the prevalence of specific topics within each genre. This genre alignment enhances interpretability, offering insights into genre-specific thematic trends.

Aggregating the gamma values (document-topic distribution) by genre will allow us to see which topics are more common within each genre.

```
# Join genre data with document-topic matrix
movie_genre_topics <- document_topics %>%
  left_join(movie_data %>% select(`Movie Name`, Genre), by = c("document" = "Movie Name"))

# Aggregate by genre and calculate average gamma per topic
genre_topic_distribution <- movie_genre_topics %>%
  group_by(Genre, topic) %>%
  summarize(avg_gamma = mean(gamma, na.rm = TRUE))
```

```
`summarise()` has grouped output by 'Genre'. You can override using the
`.groups` argument.
```

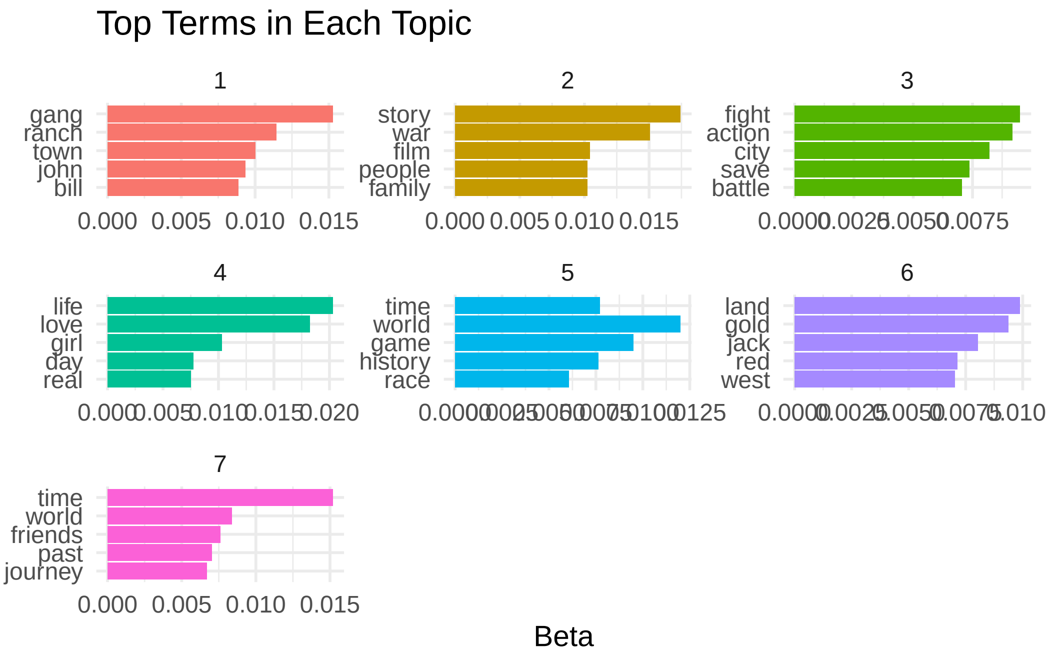
```
# Visualize topic prevalence by genre
ggplot(genre_topic_distribution, aes(x = factor(topic), y = avg_gamma, fill = Genre)) +
  geom_col(position = "dodge") +
  labs(title = "Average Topic Distribution by Genre",
       x = "Topic", y = "Average Gamma") +
  theme_minimal()
```



Top Terms per Topic with Genre Context

By examining top terms for each topic, we assign intuitive labels (e.g., "Action," "Romance") to aid in interpretation. Visualizing these terms alongside genre data provides context and supports accurate topic naming.

```r
# Visualize top terms per topic
top_terms <- topic_terms %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

ggplot(top_terms, aes(reorder(term, beta), beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  labs(title = "Top Terms in Each Topic",
       x = NULL, y = "Beta") +
  theme_minimal()
```



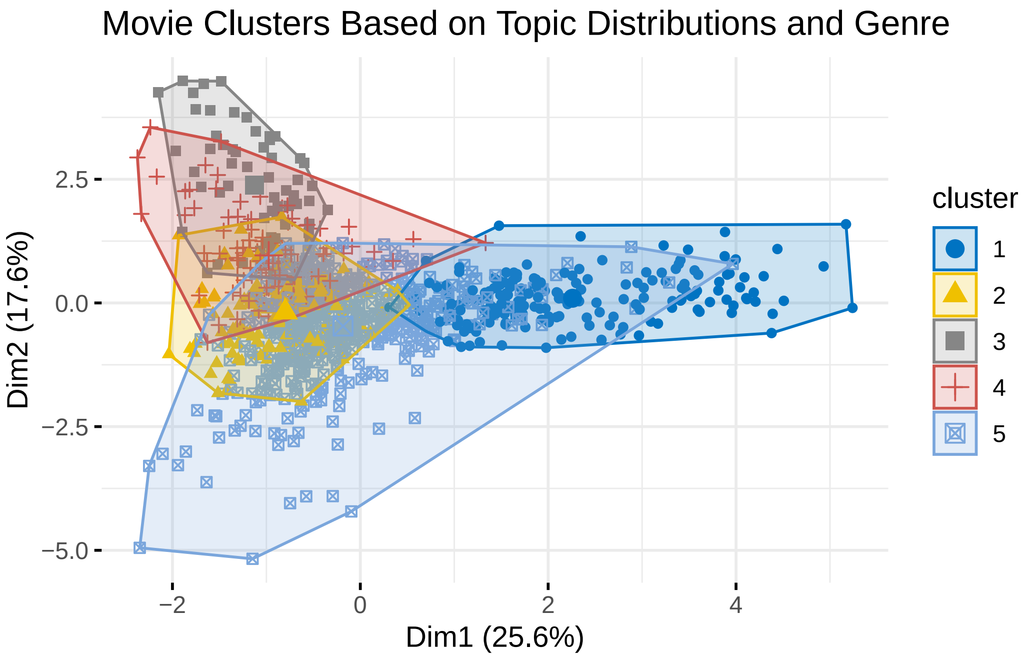
Clustering Movies by Topic with Genre Information

K-means Clustering

Applying k-means clustering to topic proportions enables us to group movies with similar thematic profiles, revealing genre clusters and potential hybrid genres. This clustering offers insights into genre connections and thematic overlap.

```r
# Reshape document-topic matrix to wide format for clustering
gamma_wide <- document_topics %>%
  pivot_wider(names_from = topic, values_from = gamma) %>%
  drop_na()

# Apply k-means clustering with 5 clusters
set.seed(1234)
clusters <- kmeans(gamma_wide %>% select(-document), centers = 5)
gamma_wide$cluster <- clusters$cluster

# Add genres to clustered data
clustered_movies <- gamma_wide %>%
  left_join(movie_data %>% select(`Movie Name`, Genre), by = c("document" = "Movie Name"))

# Visualize clusters with only convex hulls and centroids
```

```r
fviz_cluster(
  clusters,
  data = gamma_wide %>% select(-document, -cluster),
  geom = "point",             # Remove individual points
  show.clust.cent = TRUE,     # Show cluster centroids only
  ellipse.type = "convex",    # Use convex hulls
  palette = "jco",            # Choose a color palette for distinction
  ggtheme = theme_minimal()   # Clean theme for simplicity
) +
  labs(title = "Movie Clusters Based on Topic Distributions and Genre")
```



This plot effectively reveals the thematic structure within the movie dataset by clustering movies based on topic distributions. Each color-coded cluster represents a distinct grouping, with Dim1 and Dim2 capturing the most significant variance in these themes (25.6% and 17.6%, respectively).

The spread and positioning of the clusters provide insights into thematic diversity. Clusters that are more isolated indicate unique genre themes with minimal crossover, suggesting that movies in these clusters may share specialized topics that are distinct from other groups. For example, the blue cluster on the right has a clear separation, implying a strong, unique thematic focus that sets it apart from other clusters.

In contrast, clusters that overlap (e.g., the yellow and red clusters) suggest shared thematic elements or genres with blended characteristics. This overlap could indicate genres that frequently incorporate similar story elements or tropes, such as action and adventure or drama and romance.

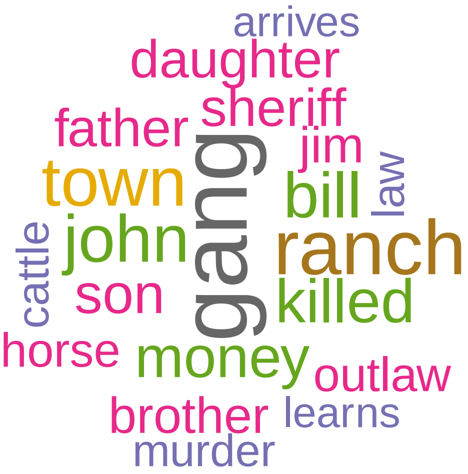
The use of convex hulls around each cluster aids in visualizing the thematic boundaries, while the distribution of points within these boundaries shows the density and variability of themes within each genre grouping. Overall, this analysis reveals both distinct genre patterns and areas of thematic overlap, providing a nuanced view of how different genres relate and diverge within the dataset.

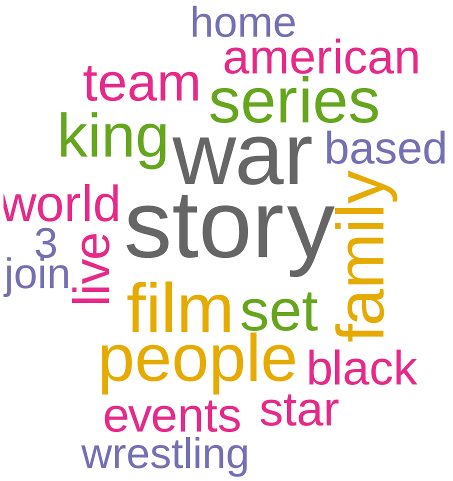
Word Clouds by Clustered Topics

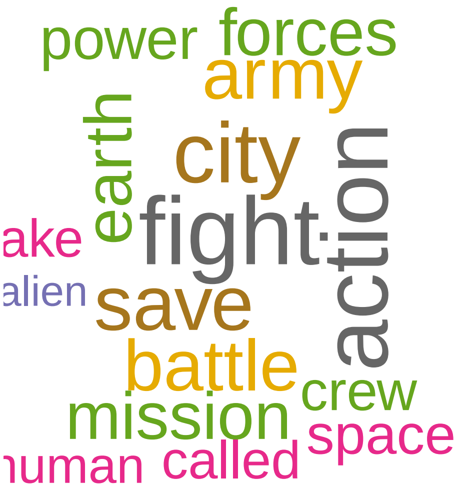
This section uses word clouds to visually highlight key terms in each topic cluster, providing an intuitive view of dominant themes across movie groups. By showcasing the most relevant words, word clouds make it easy to identify core concepts that define each cluster, offering a quick way to interpret thematic patterns in the dataset.

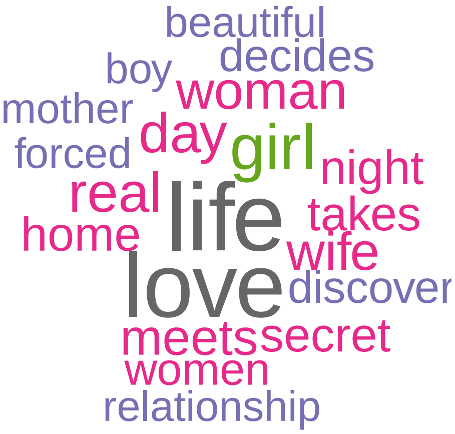
```
# Function to generate word clouds with adjusted font size and margins
create_wordcloud <- function(cluster_number) {
  terms <- topic_terms %>%
    filter(topic == cluster_number) %>%
    arrange(desc(beta)) %>%
    top_n(20, beta)

  # Scale beta values for word cloud sizes and ensure no NA values
  term_freq <- scales::rescale(terms$beta, to = c(1, 3), na.rm = TRUE)

  # Set margins and generate word cloud with specific scale
  par(mar = c(1, 1, 1, 1))  # Increase margin to prevent words from being cut off
  wordcloud(
    words = terms$term,
    freq = term_freq,
    min.freq = 0.01,
    max.words = 20,
    random.order = FALSE,
    colors = brewer.pal(8, "Dark2"),
    scale = c(3, 0.5)  # Adjust scale for smaller font size range
  )
}

# Adjust margins and layout for better visibility
par(mfrow = c(1, 2))
```

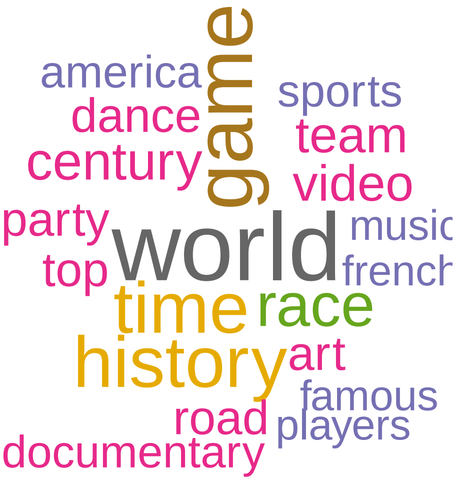
```
# Generate word clouds for each topic in the model
for (i in 1:optimal_k) {
  create_wordcloud(i)
}
```

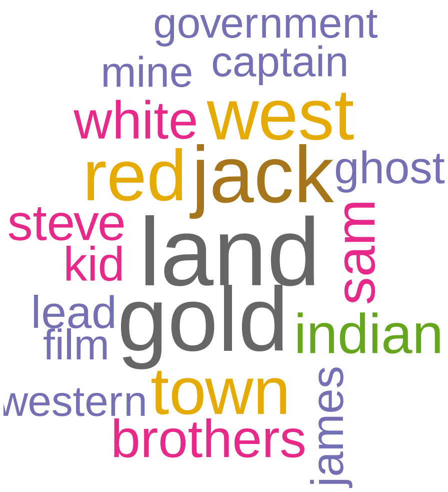


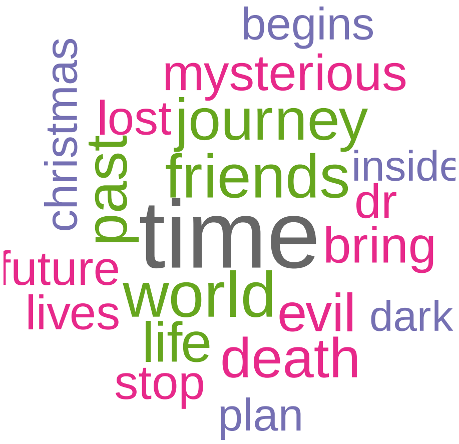







america
game
sports
dance
team
century
video
party
world
music
top
french
time
race
history
art
famous
road
players
documentary


government
mine
captain
white
west
red
jack
ghost
steve
land
sam
kid
lead
gold
indian
film
western
town
brothers
james


begins
christmas
mysterious
lost
journey
past
friends
inside
dr
time
bring
future
world
lives
evil
dark
life
death
stop
plan