Movie Plot

Dayu Tie

2024-11-07

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────── tidy
verse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────
──── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
me errors
```

```
library(tm)
```

```
## 载入需要的程序包：NLP
##
## 载入程序包：'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(topicmodels)
library(ldatuning)
library(tidytext)
library(Rtsne)
library(ggplot2)
library(wordcloud)
```

```
## 载入需要的程序包：RColorBrewer
```

```
library(RColorBrewer)
```

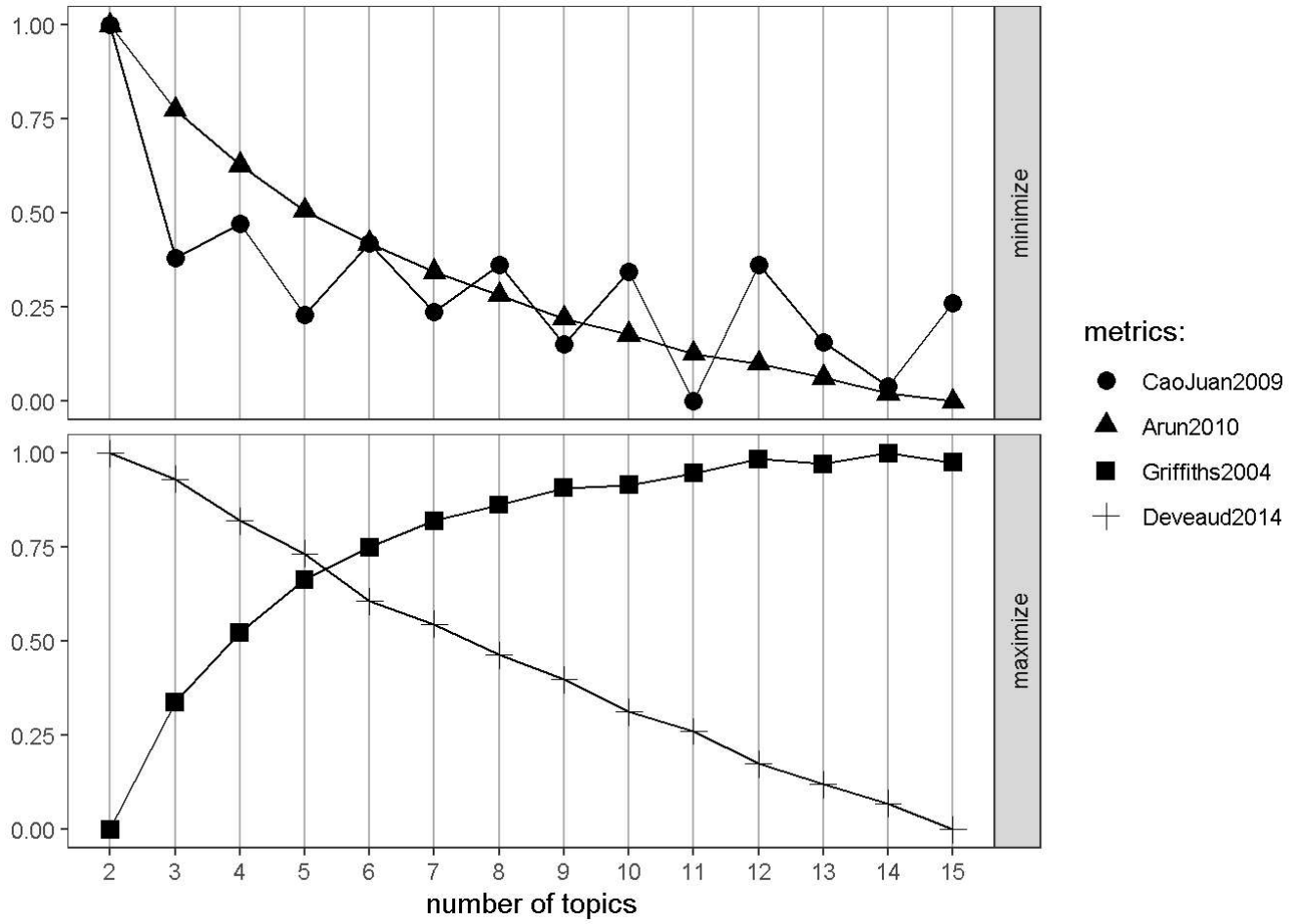
```
movie_data <- read_csv("movie_plots.csv", show_col_types = FALSE)
custom_stopwords <- c(stopwords("english"), "will", "would", "can", "could", "may", "might", "m
ust", "shall", "should", "do", "did", "does","one", "two", "three", "four", "five", "six", "sev
en", "eight", "nine", "ten","first", "second", "third", "fourth", "fifth")

corpus <- VCorpus(VectorSource(movie_data$Plot))

corpus <- corpus %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removePunctuation) %>%
    tm_map(removeWords, custom_stopwords) %>%
    tm_map(stripWhitespace)

dtm <- DocumentTermMatrix(corpus)
dtm_tfidf <- DocumentTermMatrix(corpus, control = list(weighting = weightTfIdf))
```

```
result <- FindTopicsNumber(
    dtm,
    topics = seq(2, 15, by = 1),
    metrics = c("CaoJuan2009", "Arun2010", "Griffiths2004", "Deveaud2014"),
    method = "Gibbs",
    control = list(seed = 1234),
    mc.cores = 1L,
    verbose = TRUE
)
```

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

```
FindTopicsNumber_plot(result)
```

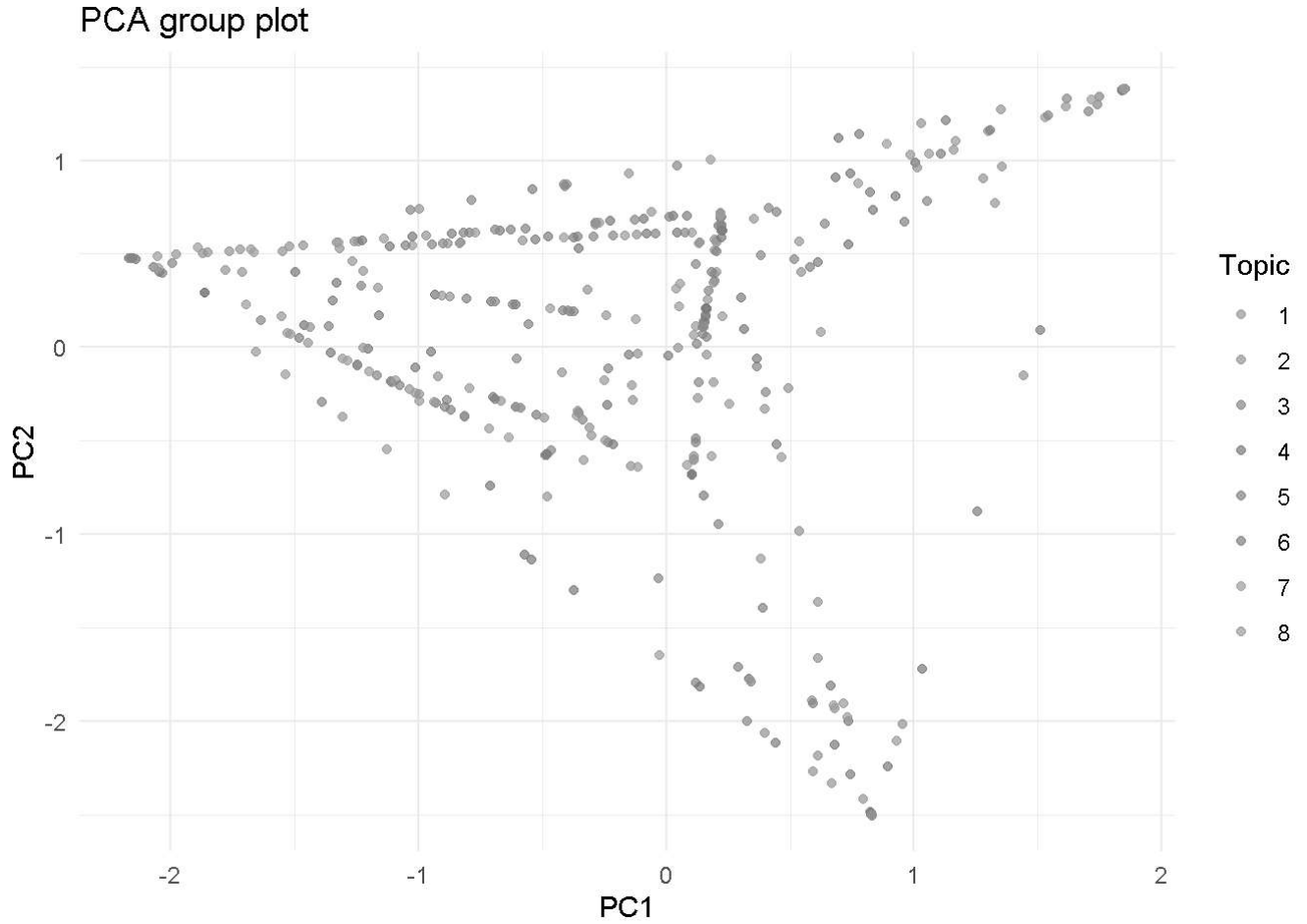
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the ldatuning package.
##   Please report the issue at <https://github.com/nikita-moor/ldatuning/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



#The scree plot helps us to observe changes in the metrics to determine the appropriate number of topics k. Often values near the inflection point can be chosen as the number of topics.CaoJuan2009 and Arun2010 (indicators to minimize): these two indicators level off after k = 5 or k = 6, suggesting that this may be a more appropriate number of topics, as continuing to increase the number of topics will not significantly improve the indicators. Griffiths2004 and Deveaud2014 (indicators to maximize): The Griffiths2004 metrics improve significantly and plateau around k = 5. The Deveaud2014 metrics plateau around k = 6. Recommended number of topics: k = 5 or k = 6 based on the above analysis.

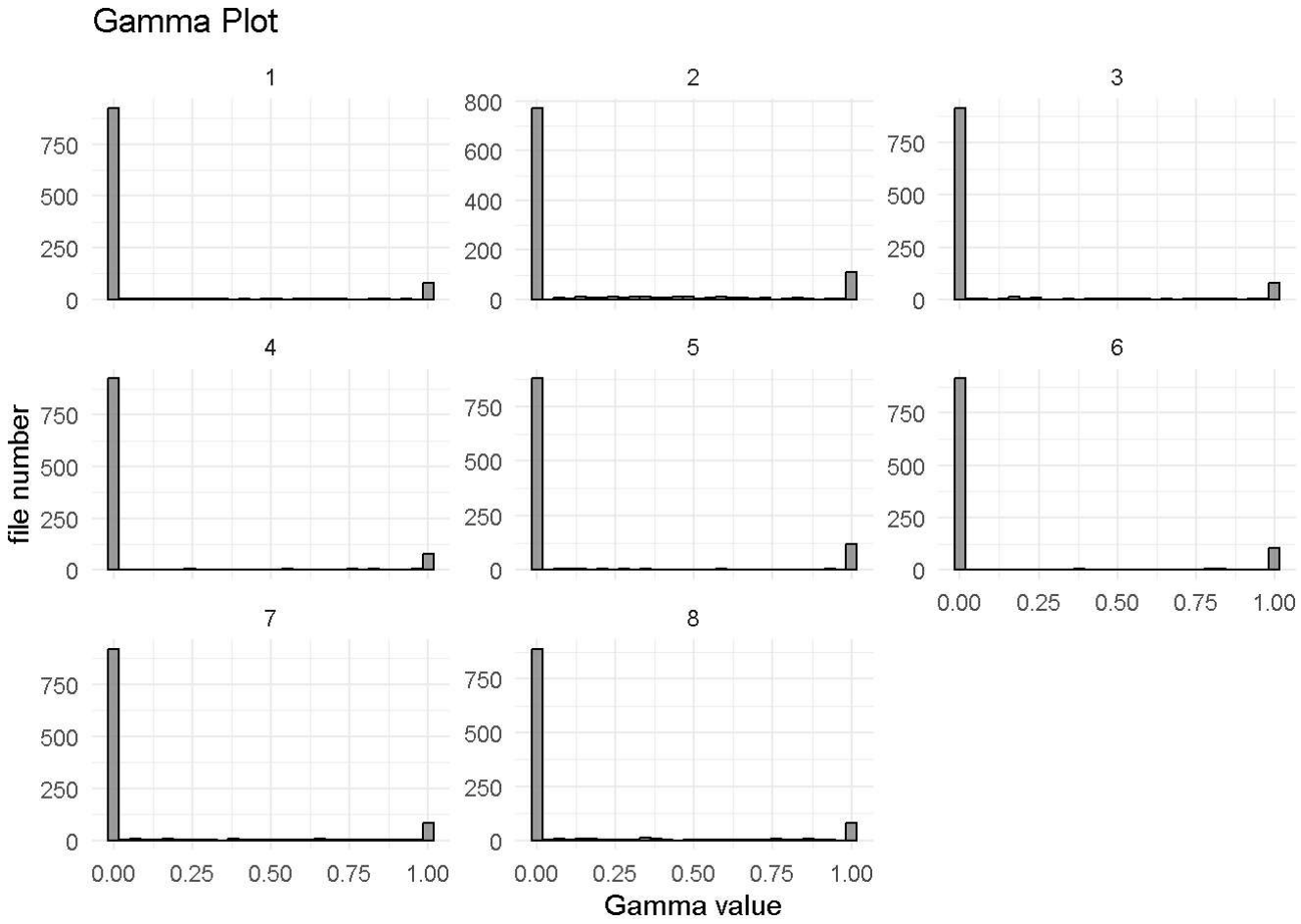
```
#set k to 6
k <- 8
lda_model<- LDA(dtm, k = 8, control = list(seed = 1234))
```

```
gamma_matrix <- posterior(lda_model)$topics

pca_model <- prcomp(gamma_matrix, center = TRUE, scale. = TRUE)
pca_data <- as.data.frame(pca_model$x)

document_topics <- tidy(lda_model, matrix = "gamma")
doc_topic <- document_topics %>%
    group_by(document) %>%
    slice_max(gamma, n = 1) %>%
    ungroup()

pca_data$Topic <- factor(doc_topic$topic)

ggplot(pca_data, aes(x = PC1, y = PC2, color = Topic)) +
    geom_point(alpha = 0.7) +
    labs(title = "PCA group plot", x = "PC1", y = "PC2") +
    theme_minimal()
```



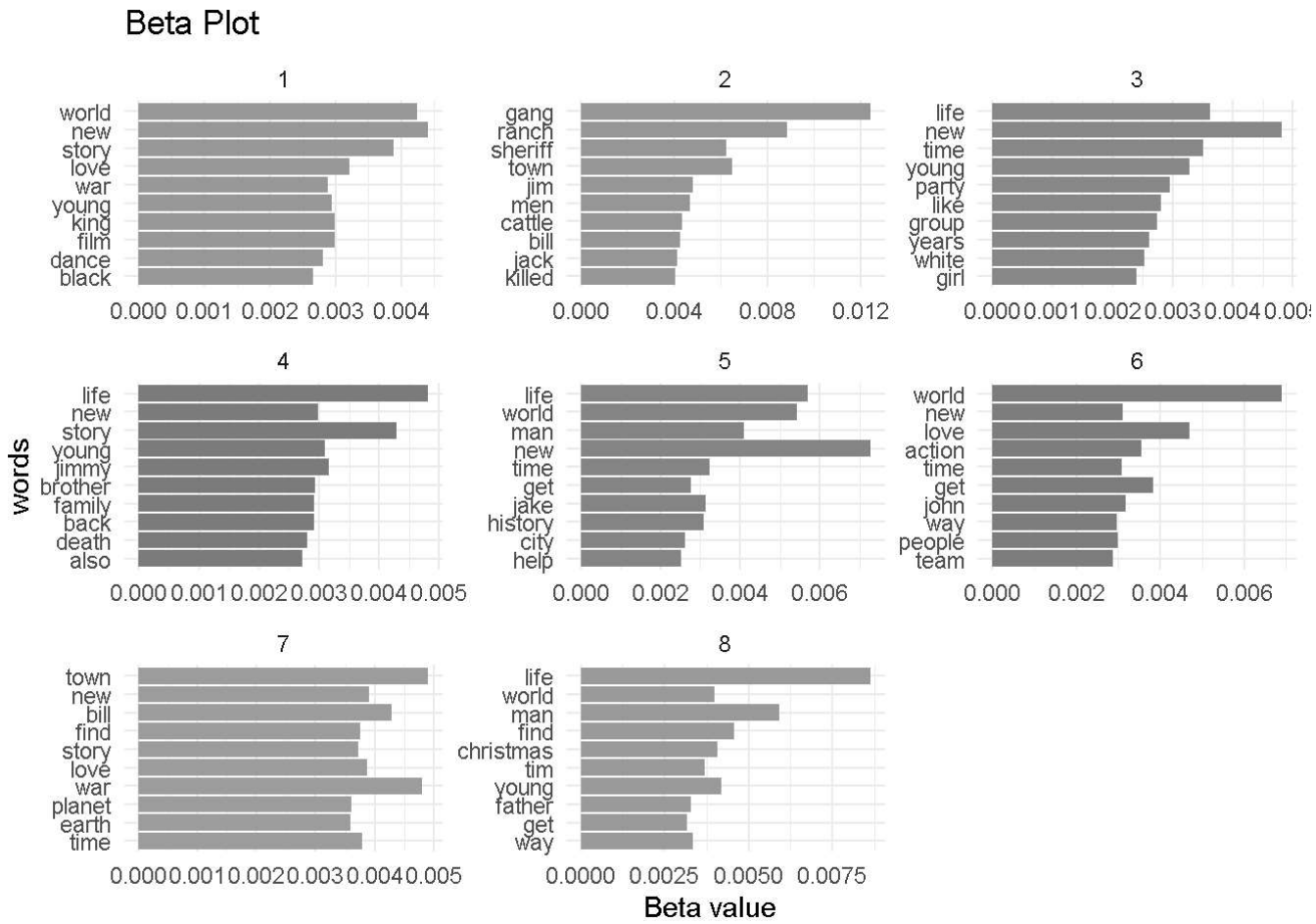
#PC1 and PC2 are the two directions that capture the greatest variability in the dataset. If a theme is more centrally distributed on one of the principal component axes, it may indicate that the content of that theme varies primarily along that direction. However, in this figure, the distribution of themes is more spread out on PC1 and PC2, indicating that these themes are not significantly linearly separated in the data.Since there are no very distinct boundaries between the themes, it can be hypothesized that there is some content overlap between the themes.

```
document_topics <- tidy(lda_model, matrix = "gamma")

# Gamma plot
ggplot(document_topics, aes(x = gamma)) +
    geom_histogram(bins = 30, fill = "steelblue", color = "black", alpha = 0.7) +
    facet_wrap(~ topic, scales = "free_y") +
    labs(title = "Gamma Plot", x = "Gamma value", y = "file number") +
    theme_minimal()
```



#The Gamma distribution shows that topic modeling has a high degree of differentiation in attributing topics to most of the documents. Documents either belong to a topic highly (Gamma close to 1) or not at all (Gamma close to 0), which implies that topic modeling is effective in classifying the topics of documents. A small number of documents have medium Gamma values, indicating that these documents have some relevance to multiple topics that may contain multiple topic features.

```
topic_terms <- tidy(lda_model, matrix = "beta")

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

ggplot(top_terms, aes(x = reorder(term, beta), y = beta, fill = factor(topic))) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    facet_wrap(~ topic, scales = "free") +
    coord_flip() +
    labs(title = "Beta Plot", x = "words", y = "Beta value") +
    theme_minimal()
```



#The Beta charts for each theme show the uniqueness of the high-frequency words that reflect the core content of each theme. Based on these high-frequency words, an initial interpretation of each theme can be made. The fact that some high-frequency words appear in more than one theme suggests that there may be some content overlap between these themes or that they are more prevalent themes in the movie.

```
palette <- brewer.pal(8, "Dark2")
all_terms <- top_terms %>%
    group_by(term) %>%
    summarize(total_beta = sum(beta)) %>%
    arrange(desc(total_beta))
wordcloud(words = all_terms$term,
          freq = all_terms$total_beta,
          min.freq = 0.001,
          colors = palette,
          random.order = FALSE,
          rot.per = 0.35,
          scale = c(3, 1))
```