2

Zhenwei Weng

2024-11-09

```
library(tidyverse)
```

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

```
library(tm)
```

```
##      NLP
##
##    'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

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

```
##      RColorBrewer
```

```
library(RColorBrewer)
```

```
movie_plot <- read_csv("movie_plots.csv", show_col_types = FALSE)

corpus <- VCorpus(VectorSource(movie_plot$Plot))
corpus_clean <- corpus %>%
  tm_map(content_transformer(tolower)) %>%
```

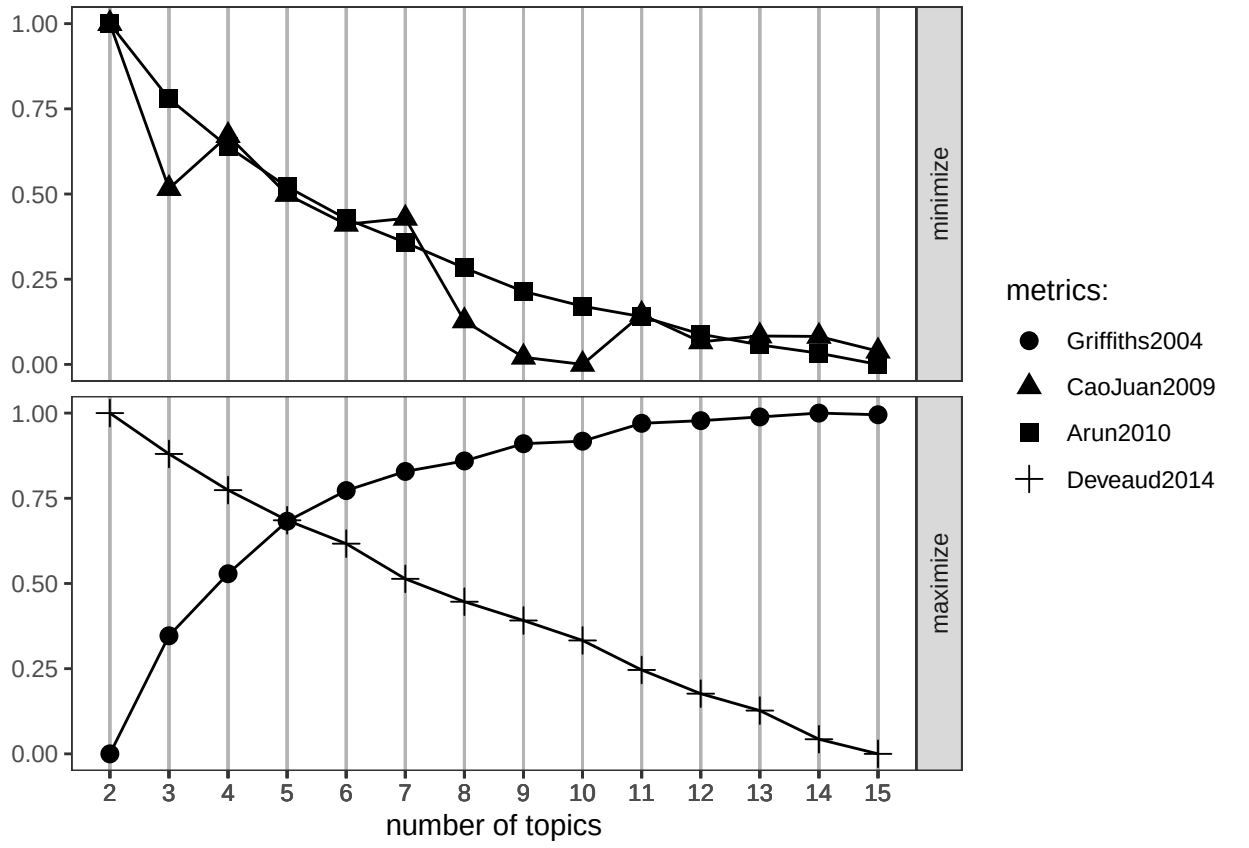
```
  tm_map(removePunctuation) %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(stripWhitespace)

dtm <- DocumentTermMatrix(corpus_clean)
```

```
# Find the best number of topics
result <- FindTopicsNumber(
  dtm,
  topics = seq(from = 2, to = 15, by = 1),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  mc.cores = 2,
  verbose = TRUE
)
```

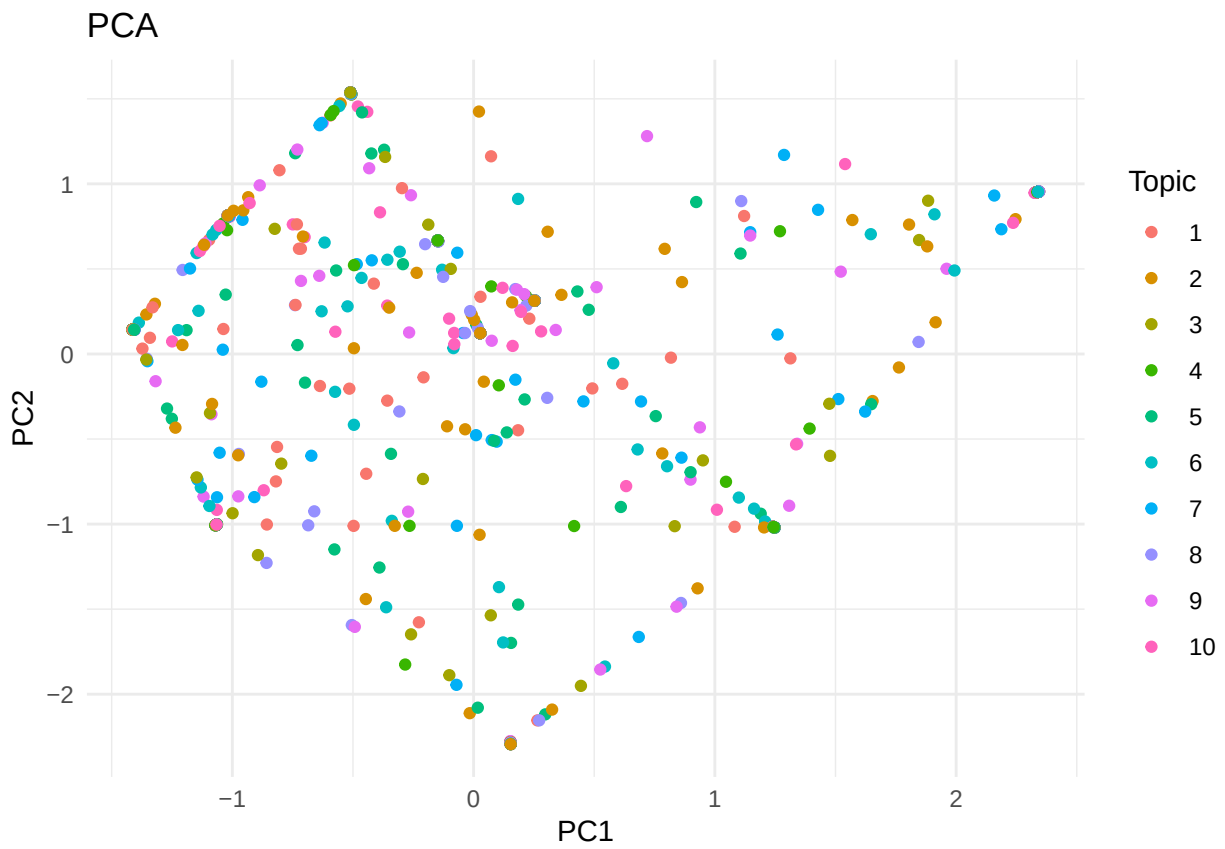
```
## fit models... done.
## calculate metrics:
##   Griffiths2004... done.
##   CaoJuan2009... done.
##   Arun2010... done.
##   Deveaud2014... done.
```

```
# Plot the metrics to help decide on the number of topics
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.
```

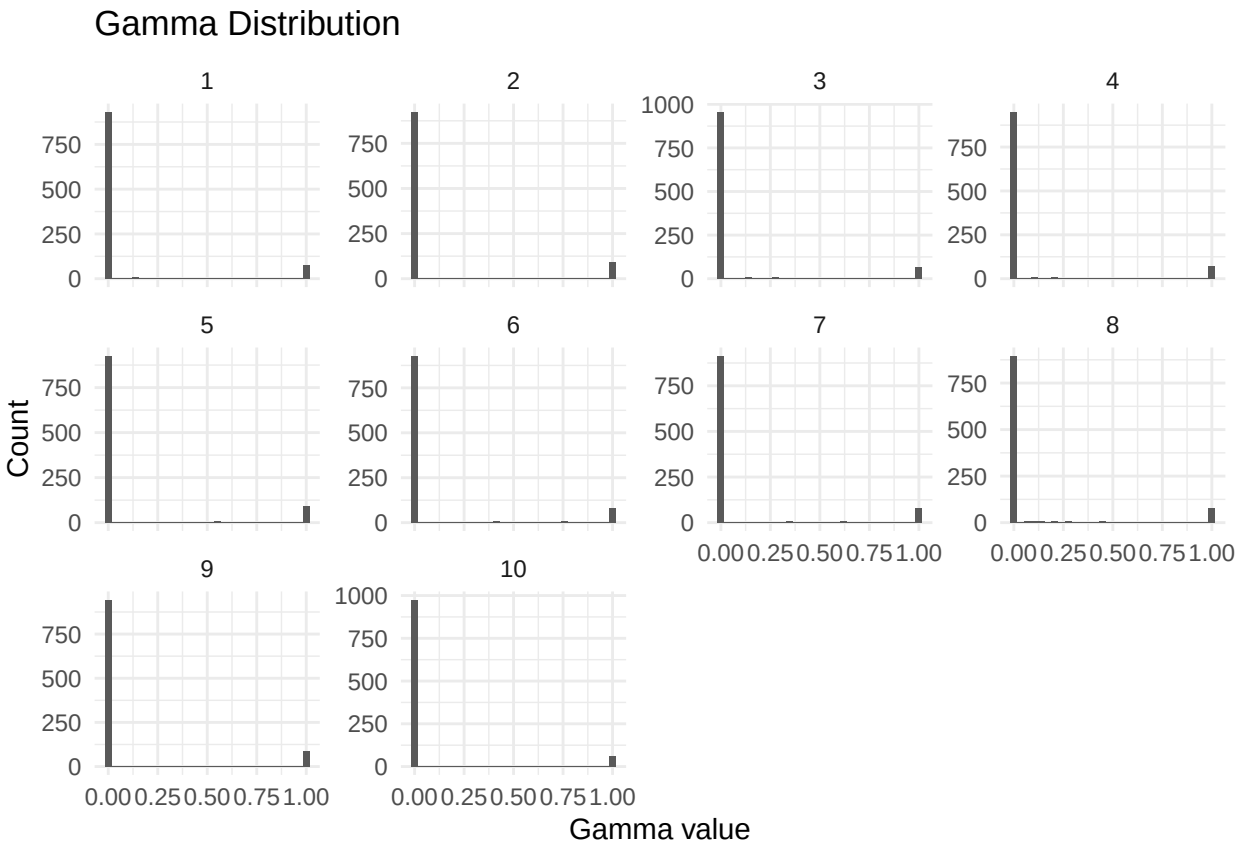



```
# Fit LDA model
num_topics <- 10  # adjust based on ldatuning results
lda_model <- LDA(dtm, k = num_topics, control = list(seed = 1234))


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(
  ) +
  labs(title = "PCA", x = "PC1", y = "PC2") +
  theme_minimal()
```



This graph visualizes whether there is a clear trend of clustering of documents across topics, which is important for evaluating the effectiveness of the model and for adjusting the number of topics.

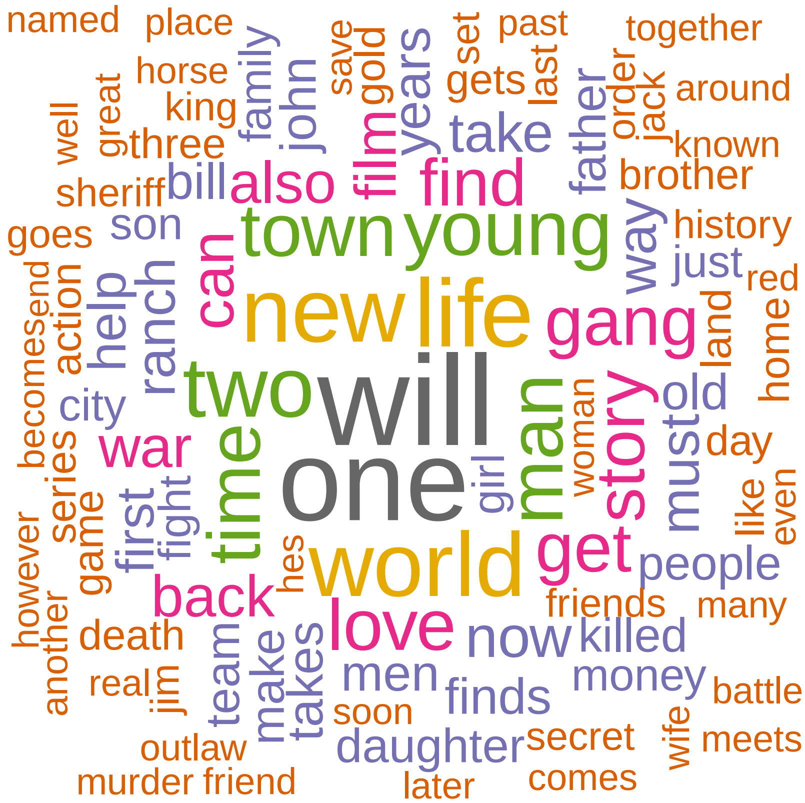
```
document_topics <- tidy(lda_model, matrix = "gamma")

# Gamma
ggplot(document_topics, aes(x = gamma)) +
    geom_histogram(bins = 30) +
    facet_wrap(~ topic, scales = "free_y") +
    labs(title = "Gamma Distribution", x = "Gamma value", y = "Count") +
    theme_minimal()
```



In this way, I can see which topics are more common in the document set and which are less common. This helps me assess the importance and impact of individual topics and allows me to further adjust the model to make it more balanced and effective.

```
# Collate terms for the word cloud based on their beta values
terms <- tidy(lda_model, matrix = "beta")

# Prepare data for the word cloud
word_cloud_data <- terms %>%
  group_by(term) %>%
  summarize(beta = sum(beta), .groups = 'drop') %>%
  arrange(desc(beta))

# Generate the word cloud
set.seed(123) # for reproducibility
par(mar=c(0,0,0,0))
wordcloud(words = word_cloud_data$term, freq = word_cloud_data$beta,
          min.freq = 1,
          max.words = 100,
          random.order = FALSE,
          rot.per = 0.35,
          colors = brewer.pal(8, "Dark2"))
```

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


will
one
new
life
world
two
town
young
time
man
gang
love
get
story
back
find
also
can
war
film
years
take
now
must
people
ranch
help
old
father
way
bill
killed
men
finds
money
daughter
city
first
fight
girl
team
make
takes
son
just
john
family
named
place
horse
king
three
great
well
sheriff
goes
end
becomes
action
series
game
however
another
death
real
jim
outlaw
murder
friend
hes
soon
later
secret
comes
save
gold
set
past
gets
last
order
jack
together
around
known
brother
history
red
land
home
woman
day
like
even
friends
many
battle
wife
meets