Topic Modeling

Group 8

2022-11-15

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidytext)
library(stringr)
library(topicmodels)
library(scales)
library(treemap)
library(tidyr)
```

```
bbc <- read.csv("bbc-news-data.csv", sep="\t", header=TRUE, stringsAsFactors = FALSE)
```

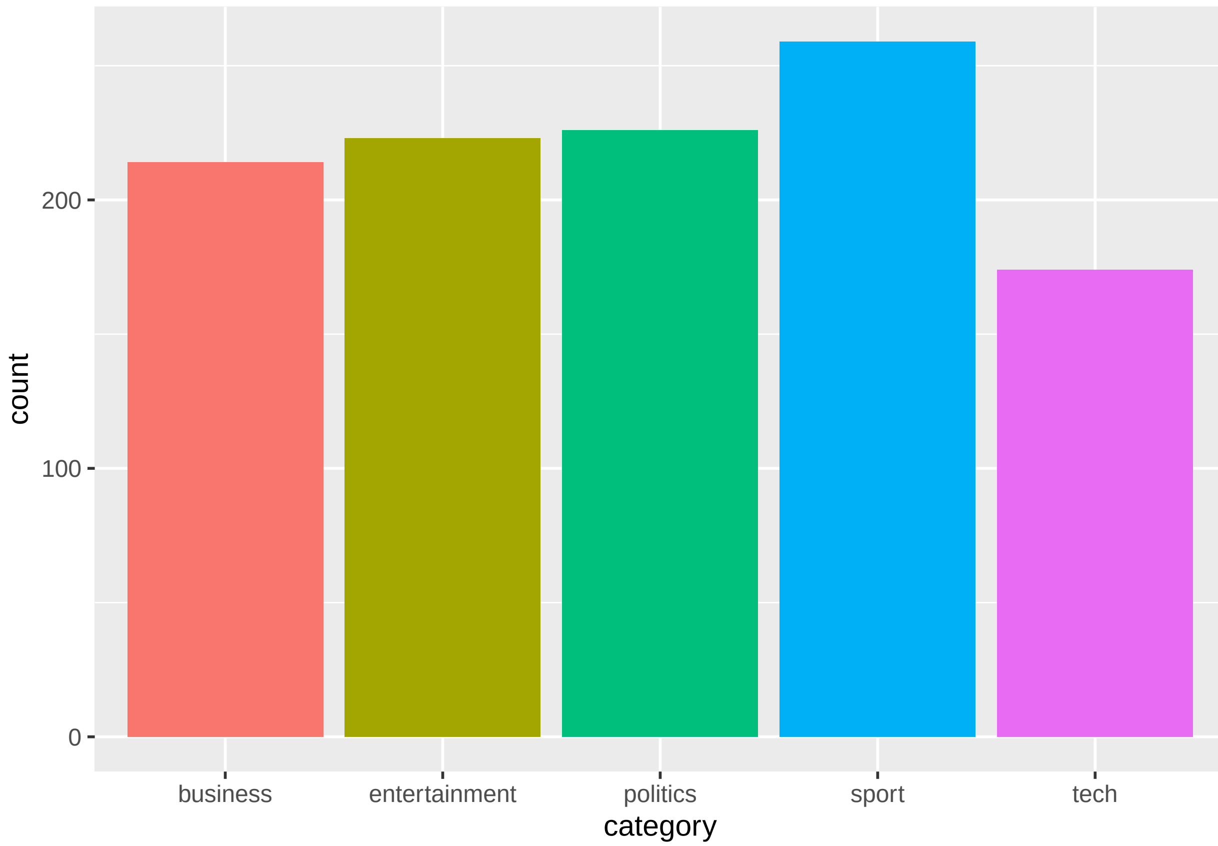
```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## EOF within quoted string
```

Description

When we get data, we need to know what it is like. So, we need to do some description about it.

```
ggplot(bbc)+
  geom_bar(aes(x=category,fill = category))+
  guides(fill=FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



We can see that sport is the most common category of news in the data set. Number of news within business, entertainment, and politics are close. Tech is the least common category.

Text mining

We now know that there are five topic news in this dataset — business, entertainment, politics, sport and tech. Let's take a closer look at what they talk about.

Word Frequency — content

We found out top 10 frequent words in contexts of each topic. It's clear that the common words vary a lot in each category, however, both politics and tech focus on the term "people".

Word Frequency — title

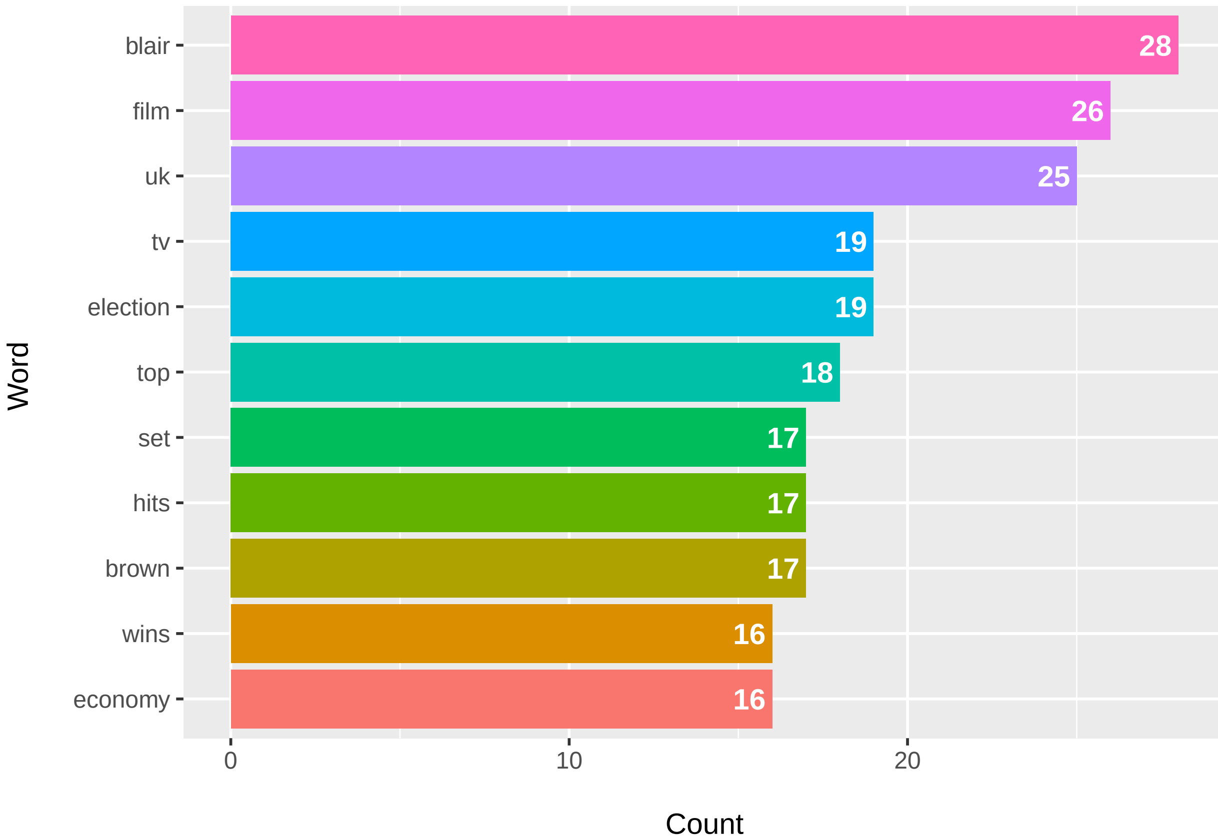
```
title_df <- data_frame(Text = bbc$title)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
title_words <- title_df %>%
  unnest_tokens(output = word, input = Text)

title_words <- title_words %>%
  anti_join(stop_words, by = "word")

title_wordcounts <- title_words %>%
  count(word, sort = TRUE)

title_wordcounts %>%
  filter(n > 15) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(aes(fill = word)) +
  coord_flip() +
```

```
  labs(x = "Word \n", y = "\n Count ") +
  geom_text(aes(label = n), hjust = 1.2, colour = "white", fontface = "bold") +
  guides(fill=FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



We summarized the most common words in news titles of all topics(words appeared more than 15 times). "Blair", "film", "UK", "TV", "election", "top", "set", "hits", "brown", "wins", and "economy" are the top eleven common words appeared in titles. Among those frequent words in titles, eight of them are highly repeated in contexts as well. "Blair", "election", and "brown" belong to the category of politics. "Wins" belongs to sport. "Economy" goes to business. Lastly, "film", "UK", and "TV" belong to entertainment. "Top", "set", and "hits" are news words that we hadn't seen in contexts.

LDA on categories

First, because 5 category in dataset, when we use LDA function, we create a five-topic model.

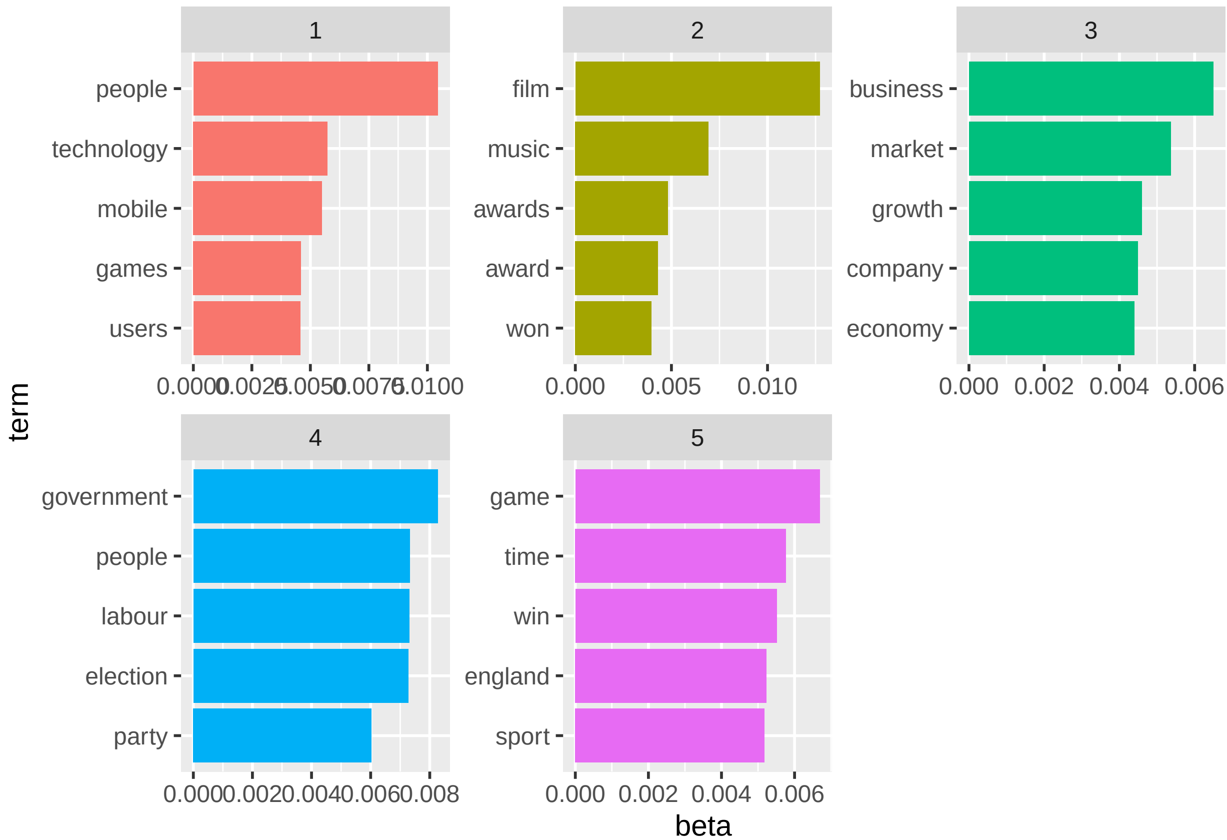
```
bbc$filename <- str_replace(bbc$filename, ".txt", "")
bbc$file <- paste(bbc$category,bbc$filename, sep = "_")
by_file_word <- bbc %>%
  unnest_tokens(word, content)
word_counts <- by_file_word %>%
  anti_join(stop_words, by = "word") %>%
  count(file, word, sort = TRUE)
file_dtm <- word_counts %>%
  cast_dtm(file, word, n)
file_lda <- LDA(file_dtm, k = 5, control = list(seed = 615))
```

Second, we try to get the probability of that term being generated from that topic. For example, the term "music" has an almost zero probability of being generated from topics 3 or 5, and it's largest probability is 0.69% in topic 2.

```
file_topics <- tidy(file_lda, matrix = "beta")
head(file_topics,10)
```

```
## # A tibble: 10 x 3
##    topic term      beta
##    <int> <chr>    <dbl>
##  1     1 music 4.44e- 3
##  2     2 music 6.92e- 3
##  3     3 music 1.15e-24
##  4     4 music 1.25e- 5
##  5     5 music 6.37e-62
##  6     1 film  8.06e- 4
##  7     2 film  1.27e- 2
##  8     3 film  4.12e- 5
##  9     4 film  4.96e-14
## 10     5 film  6.13e-23
```

```
top_terms <- file_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
top_terms
```

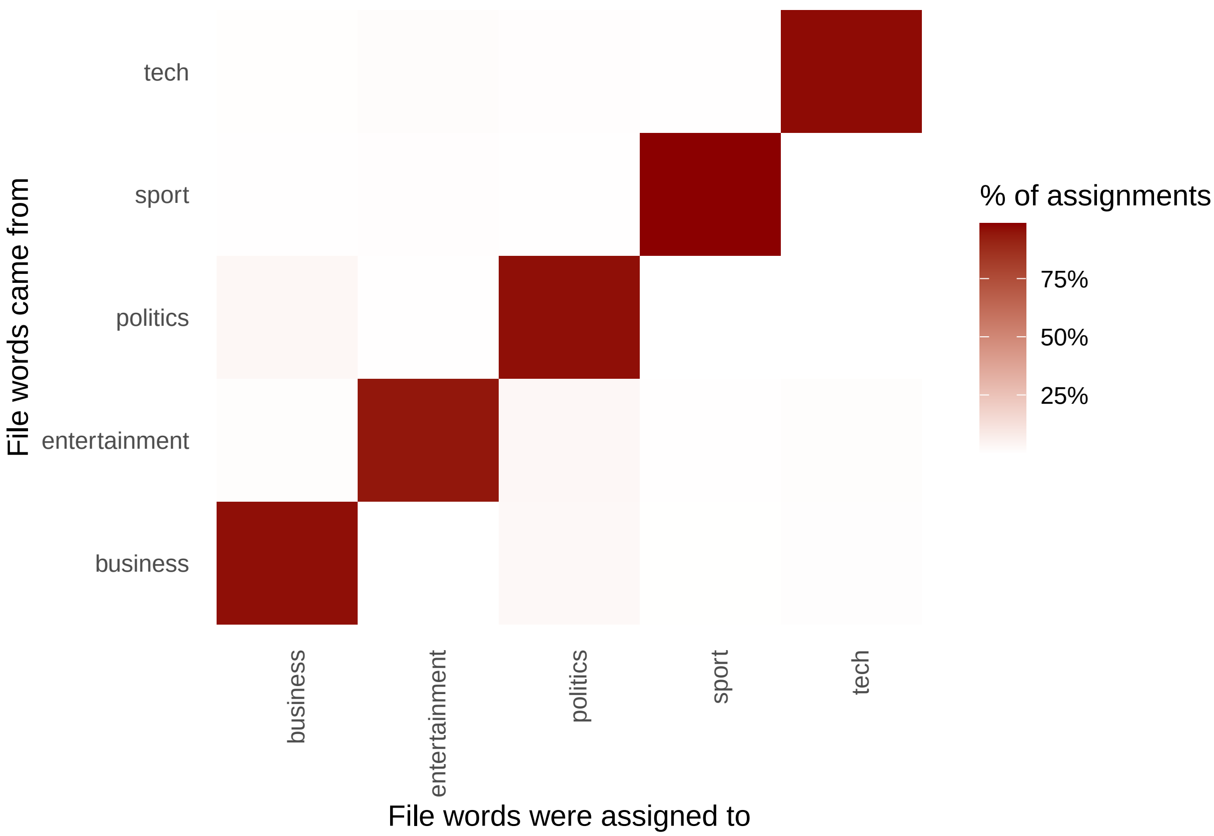


From the plot, it's clearly show that the term of "technology" and "mobile" belongs to tech, so 1 represent "tech" topic; "film" and "music" belongs to entertainment, so 2 represent "entertainment" topic; "business" ,"market" and "company" belongs to business, so 3 represent "business" topic; "government", "labour" and "election" belongs to politics, so 4 represent "politics" topic; "game" and "sport" belongs to sport, so 5 represent "sport" topic.

First, we try to reallocate the content word into five topics in order to find which words were incorrectly classified.

```
lda_gamma <- tidy(file_lda, matrix = "gamma")
lda_gamma <- lda_gamma %>%
  separate(document, c("category", "file_name"), sep = "_", convert = TRUE)
lda_classifications <- lda_gamma %>%
  group_by(category, file_name) %>%
  slice_max(gamma) %>%
  ungroup()
news_topics <- lda_classifications %>%
  count(category, topic) %>%
  group_by(category) %>%
  slice_max(n, n = 1) %>%
  ungroup() %>%
  transmute(consensus = category, topic)

assignments <- augment(file_lda, data = file_dtm)

assignments <- assignments %>%
```

```
  separate(document, c("file_title", "file_number"),
           sep = "_", convert = TRUE) %>%
  inner_join(news_topics, by = c(".topic" = "topic"))

assignments %>%
  count(file_title, consensus, wt = count) %>%
  mutate(across(c(file_title, consensus), ~str_wrap(., 20))) %>%
  group_by(file_title) %>%
  mutate(percent = n / sum(n)) %>%
  ggplot(aes(consensus, file_title, fill = percent)) +
  geom_tile() +
  scale_fill_gradient2(high = "darkred", label = percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        panel.grid = element_blank()) +
  labs(x = "File words were assigned to",
       y = "File words came from",
       fill = "% of assignments")
```



According to the picture, we can see that almost all the words for these five topics were correctly assigned, while some words in business topic had a fair number of misassigned words to politics, some words in entertainment topic had a fair number of misassigned words to politics and some words in politics topic had a fair number of misassigned words to business.

Then, let's find what were the most commonly mistaken words?

```r
wrong_words <- assignments %>%
  filter(file_title != consensus)
wrong_words %>%
  count(file_title, consensus, term, wt = count) %>%
  ungroup() %>%
  arrange(desc(n))
```

```
## # A tibble: 7,945 x 4
##    file_title    consensus     term           n
##    <chr>         <chr>         <chr>      <dbl>
##  1 politics      business      economy       39
##  2 politics      business      budget        33
##  3 politics      business      business      31
##  4 entertainment politics      government    26
##  5 tech          entertainment awards        25
##  6 tech          politics      government    25
##  7 entertainment tech          digital       24
##  8 politics      business      economic      22
##  9 politics      business      china         21
## 10 tech          entertainment playing       21
## # ... with 7,935 more rows
## # i Use `print(n = ...)` to see more rows
```

From this table, we can see that a number of words from politics topic were often assigned to business topic which match the picture above. So, we need to try some other models to find a better way to assign these words. Actually, LDA algorithm is useful but stochastic, and it can accidentally land on a topic that spans multiple files.