Quiz_3

Nadezhda Gesheva

24 Feb 2018

Problem 1

LDA is a topic modeling technique that refers to each document as a blend of various topics, while each topic is regarded as a blend of words. This technique is especially useful if we apply it to several documents where we don't know the initial number of distinct topics. Although we might refer to distinct documents, the document's content might overlap if the topics are similar.

Problem 2

My understanding of a full tidy-text analysis consists of:

- Split the text into one word-per-row dataset.
- Exclude the most common words that bring us no new information (i.e., stop words).
- Count the times that each word is contained within a text and plot the most frequent of them.
- Set the overall sentiment of the text by referring to the most frequently used words and comparing them to lexicons. We could use unigrams, bigrams, etc.
- Get the most important words within a document containing several texts, i.e. get the "uniqueness" of the document by employing tf-idf methods.
- Extract topics per chapter within a document.

Of course, in addition we could train a model that could predict whether a certain line of text belongs to certain author.

Problem 3

The topic extractor technique might be disadvantageous since often we don't know the exact number of topics that we want to determine within several text files. Therefore, we might need to make an assumption that might not be the most optimal one and make our results less robust. Nevertheless, it is very powerful technique to quickly get the topic of a large amount of text data.

Let's imagine I have the following challenge. My 3 year old daughter wants me to read her a tale from Brother Grimm's tales. She wants the tale to contain "nature" elements and other than that I am free to choose whichever I want. It's great that my kid is still interested in tales, right? I don't quite remember them all, so I have to make a quick n dirty topic modelling code so that I can determine which chapters have nature in them.

Data prep

```
# THE BROTHERS GRIMM FAIRY TALES
#install.packages("gutenbergr")
library(gutenbergr)
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(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 2.2.1     v readr   1.1.1
## v tibble  1.4.2     v purrr   0.2.4
## v tidyr   0.8.0     v stringr 1.3.0
## v ggplot2 2.2.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
#install.packages("stringr")
library(stringr)
#install.packages("topicmodels")
library(topicmodels)


titles <- c("Grimms' Fairy Tales")

grimm <- gutenberg_works(title %in% titles) %>%
  gutenberg_download(meta_fields = "title")
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

```
# remove emply lines/paragraphs
no_empty_lines <- grimm %>%
  filter(str_detect(text, regex("[:alnum:]")))

# add lines
grimm_l <- no_empty_lines %>%
  mutate(line_number = row_number())

# filter to only get the tales
grimm_tales <- grimm_l %>%
  filter(line_number > 75)

# Separate by chapters - where the title of each chapter is in UPPER CASE
by_chapter <- grimm_tales %>%
  mutate(chapter = cumsum(as.numeric(toupper(text)== text & text != '""'))) %>%
  unite(document, title, chapter)

# great!!!
# We can start topic modelling.
```

LDA modeling

```r
# Split per words
by_chapter_word <- by_chapter %>%
 unnest_tokens(word, text)

# chapter-word counts
count_words <- by_chapter_word %>%
   anti_join(stop_words) %>%
   count(document, word, sort = TRUE) %>%
   ungroup()
```

```
## Joining, by = "word"
```

```r
# necessary steps in order for tm to be installed!
#install.packages("pacman")
#pacman::p_load(tm)
# https://datascience.stackexchange.com/questions/13759/getting-error-in-rstudio-while-loading-a-packag

# transform to document term matrix format
chap_doc_term_mat <- count_words %>%
  cast_dtm(document, word, n)

# Assumption - let k = 6. We have more than 60 chapters, and I assume that there must be
# at least 6 different topics among them.
chapters_lda <- LDA(chap_doc_term_mat, k = 6, control = list(seed = 1234))
chapters_lda
```

```
## A LDA_VEM topic model with 6 topics.
```

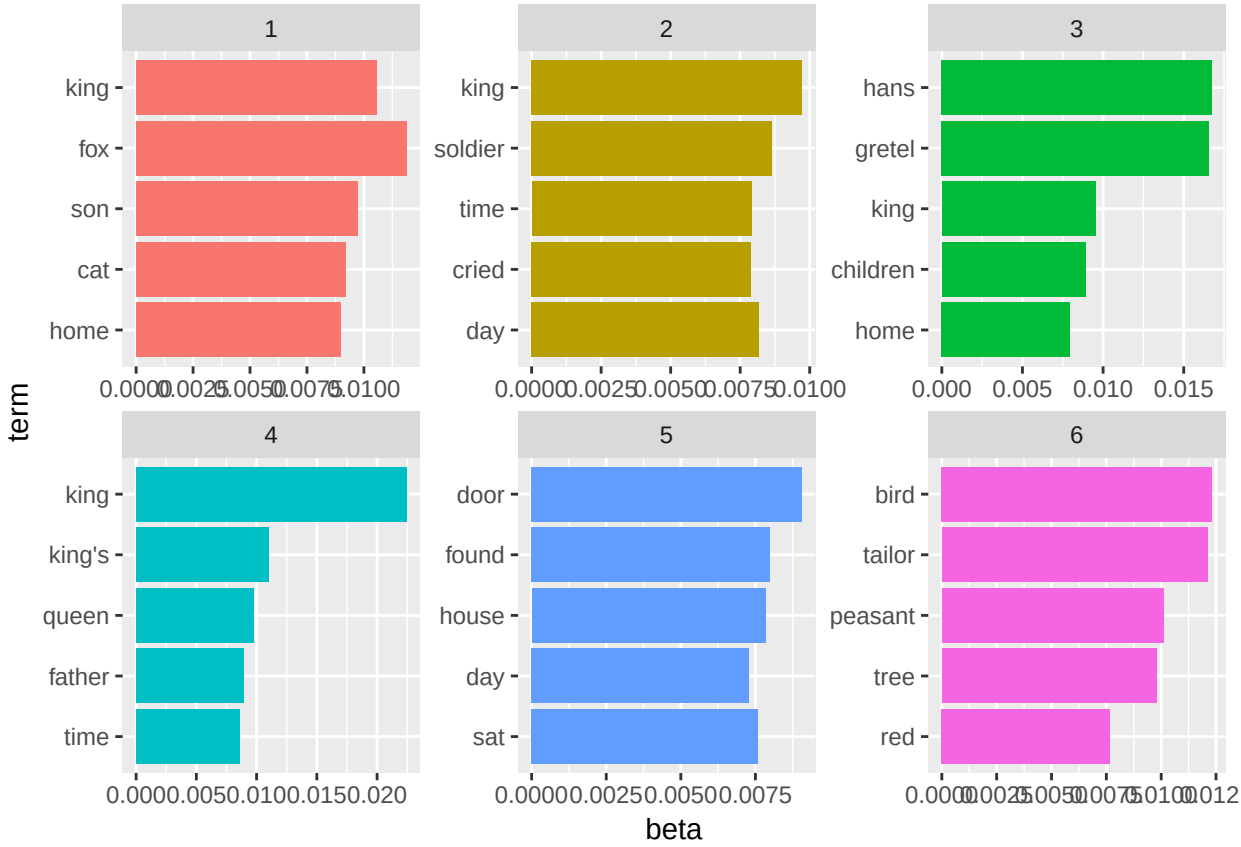
```r
# per-topic per-word probabilities
chapter_topics <- tidy(chapters_lda, matrix = "beta")
chapter_topics
```

```
## # A tibble: 26,238 x 3
##    topic term                                                      beta
##    <int> <chr>                                                    <dbl>
##  1     1 hans                                                 5.09e-193
##  2     2 hans                                                 3.89e-190
##  3     3 hans                                                 1.67e-  2
##  4     4 hans                                                 2.35e-  3
##  5     5 hans                                                 3.46e-  3
##  6     6 hans                                                 1.24e-192
##  7     1 tailor                                               3.65e-191
##  8     2 tailor                                               2.79e-190
##  9     3 tailor                                               1.05e-  3
## 10     4 tailor                                               2.36e- 18
## # ... with 26,228 more rows
```

```r
# get the top words per topic
top_terms <- chapter_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

#top_terms
```

```
# Vizualize
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```




```
# get the probability of each topic existing in each chapter
chapters_gamma <- tidy(chapters_lda, matrix = "gamma")
# Topic 6 contains the word "bird" and "tree" with high probability!
# This qualifies as "nature".

# See top 10 chapters with topic 6 that contains bird and tree
chapters_gamma %>%
  filter(topic == 6) %>%
  arrange(desc(gamma)) %>%
  top_n(10)
```

```
## Selecting by gamma
```

```
## # A tibble: 10 x 3
##    document                topic gamma
##    <chr>                   <int> <dbl>
##  1 Grimms' Fairy Tales_43      6 1.000
##  2 Grimms' Fairy Tales_31      6 1.000
##  3 Grimms' Fairy Tales_62      6 1.000
```

```
##  4 Grimms' Fairy Tales_25     6 1.000
##  5 Grimms' Fairy Tales_44     6 1.000
##  6 Grimms' Fairy Tales_37     6 1.000
##  7 Grimms' Fairy Tales_23     6 1.000
##  8 Grimms' Fairy Tales_57     6 1.000
##  9 Grimms' Fairy Tales_42     6 1.000
## 10 Grimms' Fairy Tales_30     6 0.999
```

```
# We are almost done! I could pick Grimms' Fairy Tales_43.

# Let's double check.
# Remove stop_words and count the most often used words by the Grimm brothers
# in chapter 43
by_chapter_word %>%
  anti_join(stop_words) %>%
  filter(document == "Grimms' Fairy Tales_43") %>%
  count(word, sort=TRUE) %>%
  top_n(10)
```

```
## Joining, by = "word"
```

```
## Selecting by n
```

```
## # A tibble: 10 x 2
##    word          n
##    <chr>     <int>
##  1 bird         39
##  2 mother       24
##  3 tree         21
##  4 wife         18
##  5 juniper      17
##  6 beautiful    16
##  7 kywitt       16
##  8 father       15
##  9 sing         15
## 10 marleen      14
```

"Long, long ago, some two thousand years or so, there lived a rich man with a good and beautiful wife. . . "