Twitter LDA Topic Modeling

🧭 Overview & Setup

This tutorial walks you through on:

1. How to preprocess text for text analysis
2. How to perform a LDA topic modeling analysis
3. How to plot the top terms for each topic

📦 Install and load packages

Install the following packages if you do not have them in your R environment.

- `tidyverse` is a collection of R packages for data science
- `tidytext` is used to preprocess data for text mining
- `topicmodels` is used to perform Latent Dirichlet Allocation (LDA) topic modeling analysis
- `reshape2` is a dependency that may need to be installed manually
- `LDAvis` is used to interactively visualize topic modeling results using a web-based viewer

Note that the result of your analysis may differ based on the `tidytext` version. Specifically, the tokenizing logic from version `0.4.0` (released on December 20th, 2022) and above has been updated. `token = "tweets"` option has been deprecated and will throw an error. This notebook assumes that you're using `tidytext` version `>=0.4.0`.

```r
# uncomment and run the lines below if you need to install these packages

# install.packages("tidyverse")
# install.packages("tidytext")
# install.packages("topicmodels")
# install.packages("reshape2")
# install.packages("LDAvis")
# install.packages("servr")
```

Load packages.

```r
library(tidyverse)
library(tidytext)
library(topicmodels)
library(reshape2)
library(LDAvis)
library(servr)
```

📃 Read CSV file

```
df_tweets = read_csv('Lululemon-tweets.csv')
```

```
## Rows: 4516 Columns: 3
## ── Column specification ──────────────────────────────────────────────────────
───
## Delimiter: ","
## chr (2): username, text
## dbl (1): id
##
## 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 mess
age.
```

```
df_tweets %>% head()
```

id <dbl>	username <chr>
1.603435e+18	demsRinsane
1.603434e+18	sherbertkuma
1.603433e+18	aidaampie
1.603433e+18	marinarodich
1.603433e+18	poshcitystore
1.603432e+18	marinarodich

6 rows | 1-2 of 3 columns

Print out the number of rows.

```
nrow(df_tweets)
```

```
## [1] 4516
```

🔨 Text Pre-processing

📌 Add row number to df_tweets

Create a new column named `row_num` with unique values in each row. This allows us to group by each tweet after we tokenize the tweet text.

```
df_tweets$row_num <- seq_len(nrow(df_tweets))
df_tweets %>% head() %>% select(-text)
```

id <dbl>	username <chr>	row_num <int>
1.603435e+18	demsRinsane	1
1.603434e+18	sherbertkuma	2
1.603433e+18	aidaampie	3
1.603433e+18	marinarodich	4
1.603433e+18	poshcitystore	5
1.603432e+18	marinarodich	6

6 rows

🔗 Remove URLs

Many tweets contain URL strings in the form of “https://t.co/somestring (https://t.co/somestring)”. Remove these URL strings using a regular expression

```
df_tweets$text <- df_tweets$text %>%
  str_remove_all("https://t.co/\\w+")

df_tweets %>%
  select(text) %>%
  head()
```

text <chr>
@GOPChairwoman YOU NEED TO RESIGN. Lululemon you smug bitch 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻 🖕🏻
i want the lululemon pink water bottle so bad LOL i've been eyeing it for a couple months now but als many water bottles already hahha
Only thing I want for Christmas is lululemon 🤭 life without my employee discount is not it
Check out this listing I just added to my #Poshmark closet: Lululemon Groove Pant SHR Flare *Nulu C Merlot Size-12 NWT. #shopmycloset @poshmarkapp
Check out this listing I just found on #Poshmark: Lululemon Base Pace High-Rise Running Tight 25" i Electric. #shopmycloset @poshmarkapp
Check out this listing I just added to my #Poshmark closet: Lululemon Groove Pant SHR Flare *Nulu C Merlot Size-12 NWT. #shopmycloset @poshmarkapp

6 rows

⚔️ Tokenize and normalize

Tokenize tweet texts and normalize the tokens.

```
# tokenize using unnest_tokens()
# this also normalizes the tokens (lowercase, remove punctuations except Twitter
-specific characters for mentions, tickers, and URLs)
# token = "tweets" preserves usernames and hashtags
df_tokens <- df_tweets %>%
  tidytext::unnest_tokens(input = text, output = word)

df_tokens %>% head(n = 10)
```

id	username	row_num	word
<dbl>	<chr>	<int>	<chr>
1.603435e+18	demsRinsane	1	gopchairwoman
1.603435e+18	demsRinsane	1	you
1.603435e+18	demsRinsane	1	need
1.603435e+18	demsRinsane	1	to
1.603435e+18	demsRinsane	1	resign
1.603435e+18	demsRinsane	1	lululemon
1.603435e+18	demsRinsane	1	you
1.603435e+18	demsRinsane	1	smug
1.603435e+18	demsRinsane	1	bitch
1.603434e+18	sherbertkuma	2	i

1-10 of 10 rows

🪓 Remove stop words

Stop words are words that are commonly used and likely unimportant. Examples include “is”, “by”, “the”, “a”, etc.

```
# remove stop words using anti_join
# and remove tokens with only 1 or 2 characters
df_tokens <- df_tokens %>%
  anti_join(tidytext::stop_words, by = "word") %>%
  filter(nchar(word) >= 3)

df_tokens %>% head(n = 10)
```

id	username	row_num	word
<dbl>	<chr>	<int>	<chr>
1.603435e+18	demsRinsane	1	gopchairwoman
1.603435e+18	demsRinsane	1	resign
1.603435e+18	demsRinsane	1	lululemon
1.603435e+18	demsRinsane	1	smug
1.603435e+18	demsRinsane	1	bitch

id <dbl>	username <chr>	row_num <int>	word <chr>
1.603434e+18	sherbertkuma	2	lululemon
1.603434e+18	sherbertkuma	2	pink
1.603434e+18	sherbertkuma	2	water
1.603434e+18	sherbertkuma	2	bottle
1.603434e+18	sherbertkuma	2	bad

1-10 of 10 rows

🔮 Most frequent tokens

```
df_tokens %>% count(word) %>% arrange(desc(n))
```

word <chr>	n <int>
lululemon	4491
check	1361
poshmark	1311
poshmarkapp	1311
shopmycloset	1310
listing	1276
closet	1222
added	1221
lulu	468
size	426

1-10 of 7,794 rows Previous **1** 2 3 4 5 6 ... 780 Next

🔨 LDA Analysis

📐 Create a Document-term Matrix

- Each row in our Document-term Matrix represents a tweet.
- Each column represents a word (e.g., “bankruptcy”).
- Each cell contains the frequency of the word.

```
dtm <- df_tokens %>%
  count(row_num, word) %>%
  cast_dtm(document = row_num, term = word, value = n)

dtm
```

```
## <<DocumentTermMatrix (documents: 4515, terms: 7794)>>
## Non-/sparse entries: 47787/35142123
## Sparsity           : 100%
## Maximal term length: 48
## Weighting          : term frequency (tf)
```

🧪 Run LDA with 3 topics (`k = 3`)

```
tweets_lda <- topicmodels::LDA(dtm, k = 3, control = list(seed = 12))
tweets_lda
```

```
## A LDA_VEM topic model with 3 topics.
```

Print out per-topic-per-word probabilities.

beta values are the probabilities of words in each topic.

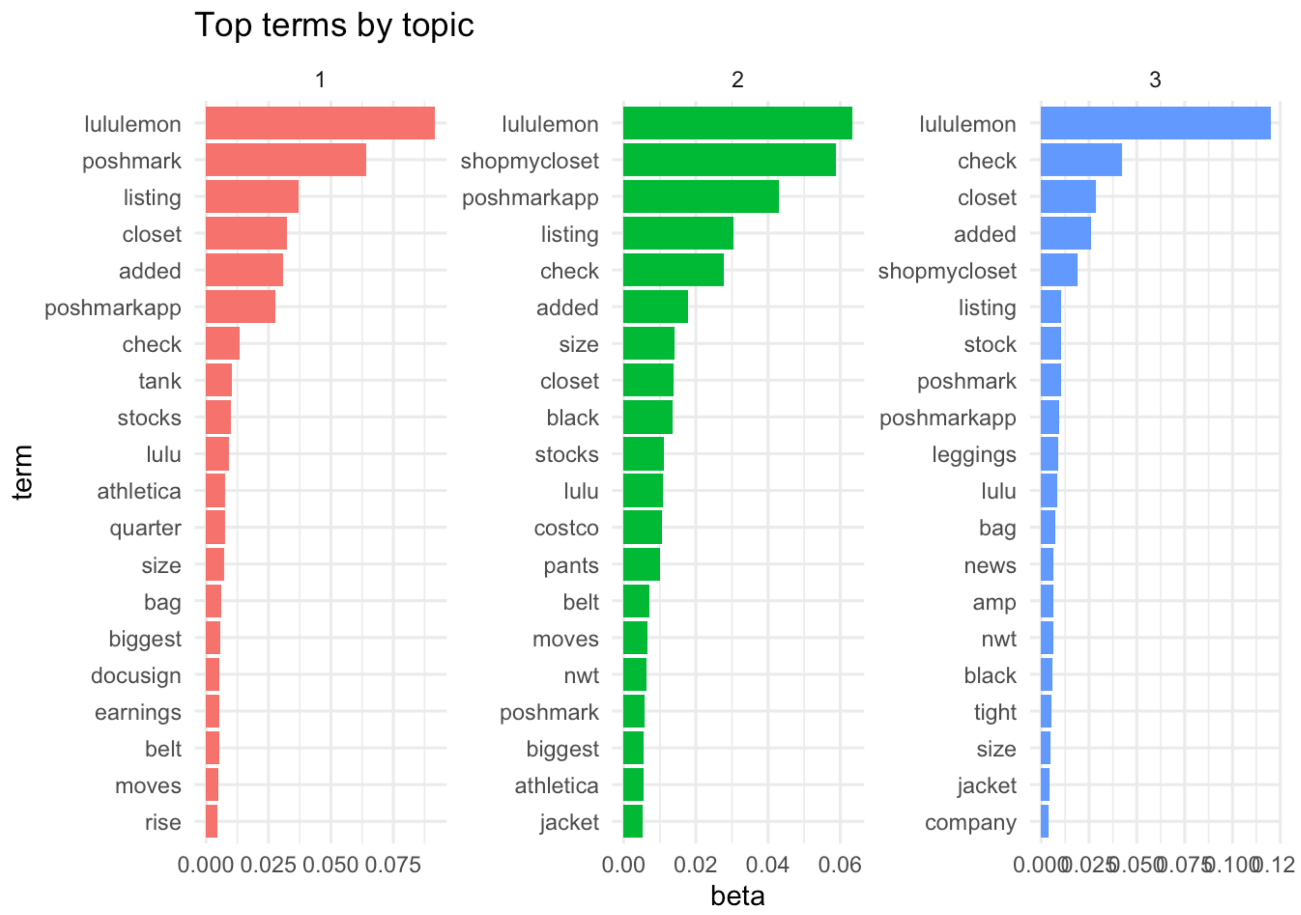
```
tweet_topics <- tidytext::tidy(tweets_lda, matrix = "beta")

top_terms <- tweet_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 20) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

topic	term	beta
<int>	<chr>	<dbl>
1	lululemon	0.091726068
1	poshmark	0.064125891
1	listing	0.037191175
1	closet	0.032374210
1	added	0.030900278
1	poshmarkapp	0.027761416
1	check	0.013568390
1	tank	0.010188724
1	stocks	0.009777794
1	lulu	0.009256346

1-10 of 60 rows Previous **1** 2 3 4 5 6 Next

📊 Plot the top 20 terms

```
top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic)))+
  geom_col(show.legend = FALSE) +
  theme_minimal() +
  ggtitle("Top terms by topic") +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered()
```



🌌 (Optional) Intertopic Distance Map

The intertopic distance map will not be shown in the knitted HTML file. Run the R notebook to see the intertopic distance map.

```
post <- topicmodels::posterior(tweets_lda)
mat <- tweets_lda@wordassignments
json <- LDAvis::createJSON(
    phi = post$terms,
    theta = post$topics,
    vocab = colnames(post$terms),
    doc.length = slam::row_sums(mat, na.rm = TRUE),
    term.frequency = slam::col_sums(mat, na.rm = TRUE)
)
serVis(json)
```