Lab Section 7: Topic Modeling in R

Napon Jatusripitak

5/17/2019

LDA using package topicmodels

Yelp Reviews

In this example, we will be fitting a topic model on yelp reviews. Our objective is to dicover latent structure in the data and discern different topics that, hopefully, correspond to distinct types of reviews in a meaningful way (ie. positive/negative, service, atmosphere, food, drinks, etc.).

Step 1: Importing and Preparing the Data for Preprocessing

```
setwd("~/Documents/GitHub/MMSS_311_2/Lab Exercises/Week 7")
df <- read.csv("review.csv", stringsAsFactors = F)
```

Step 1: Importing and Preparing the Data for Preprocessing

```
df <- df %>%
  mutate(doc_id = row_number()) %>%
  select(doc_id, text, everything()) %>%
  sample_n(size = 20000)
```

Step 2: Preprocessing with tm

Creating a corpus

```r
corpus <- VCorpus(DataframeSource(df)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeWords, stopwords('english')) %>%
  tm_map(stemDocument) %>%
  tm_map(stripWhitespace)
```

Step 2: Preprocessing with tm

Creating a Document-Term Matrix

```r
dtm <- corpus %>%
  DocumentTermMatrix() %>%
  removeSparseTerms(sparse = 0.99)
```

LDA

```r
mod.out.5 <- LDA(dtm, k=5, control = list(seed=6))
```

```
## Error in LDA(dtm, k = 5, control = list(seed = 6)): Each
```

Step 3: Preparing the Data for Topic Modeling (aka. Removing Empty Rows)

Option A (https://stackoverflow.com/questions/13944252/remove-empty-documents-from-documenttermmatrix-in-r-topicm

1. Sum by row and store the result in a vector
2. Using this vector, retain only the rows from the document-term matrix that have this sum greater than 0.

```
#Find the sum of words by row
rowTotals <- apply(dtm, 1, sum)
#Remove all docs without words
dtm   <- dtm[rowTotals> 0, ]
```

Step 3: Preparing the Data for Topic Modeling (aka. Removing Empty Rows)

Option B

1. Sum by row and store the rows that have this sum equal to 0 in a vector
2. Using this vector, modify the corpus to retain only the rows that have the sum greater than 0
3. Recreate the document-term matrix using the newly modified corpus

```r
empty_rows <- which(rowSums(as.matrix(dtm)) == 0)
dtm <- corpus[-empty_rows] %>%
  DocumentTermMatrix() %>%
removeSparseTerms(sparse = 0.99)
```

Step 4: Fitting a Topic Model (LDA)

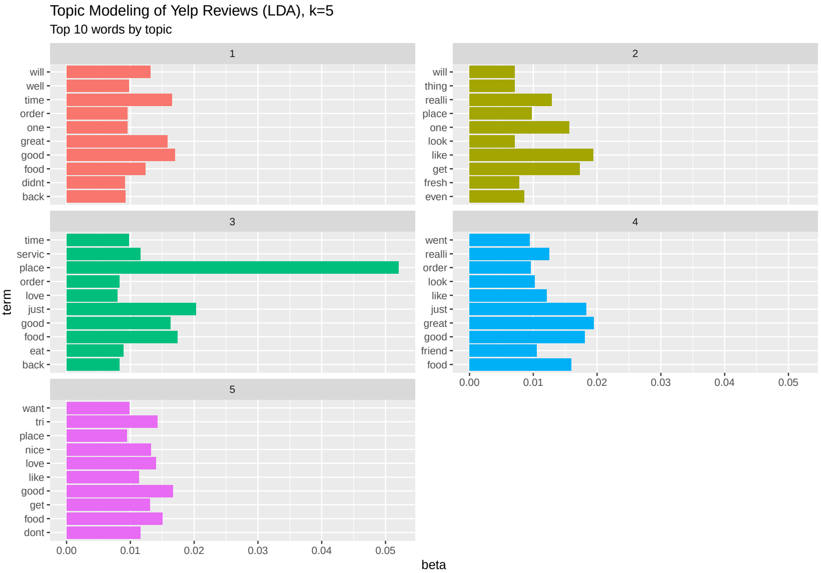
```
mod.out.5 <- LDA(dtm, k=5, control = list(seed=6))
```

Step 5: Visualization

Get Top Terms

```r
tidy(mod.out.5) %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y", nrow = 4) +
  coord_flip() +
  xlab("term") +
  labs(title = ’Topic Modeling of Yelp Reviews (LDA), k=5’,
  subtitle = ’Top 10 words by topic’)
```

Step 5: Visualization



Determining the optimal number of topics

```r
perplexity(mod.out.5)
```

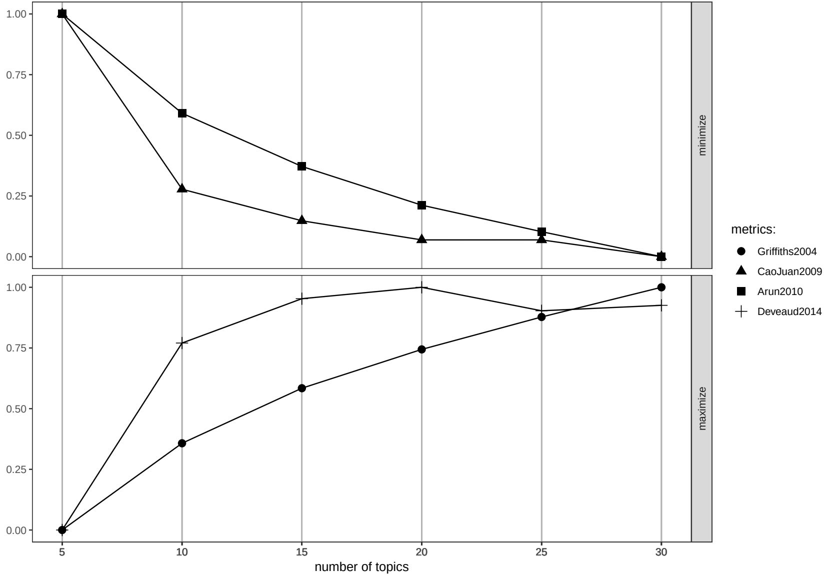
```
## [1] 586.2251
```

Determining the optimal number of topics

```
result <- FindTopicsNumber(
  dtm,
  topics = seq(from = 5, to = 30, by = 5),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Devea
  method = "Gibbs",
  control = list(seed = 77),
  mc.cores = 2L,
  verbose = TRUE
)
```

Determining the optimal number of topics

```
FindTopicsNumber_plot(result)
```



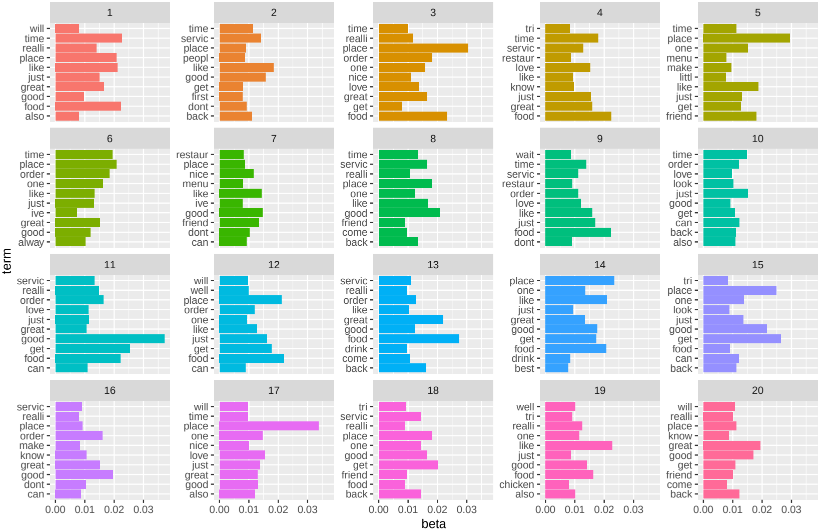
Refitting the model with k=20

```
mod.out.20 <- LDA(dtm, k=20, control = list(seed=77))
```

Visualization

```
tidy(mod.out.20) %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y", nrow = 4) +
  coord_flip() +
  xlab("term") +
  labs(title = ’Topic Modeling of Yelp Reviews (LDA), k=20’,
  subtitle = ’Top 10 words by topic’)
```

Visualization



Topic Modeling of Yelp Reviews (LDA), k=20

Top 10 words by topic

Topic Modeling with stm

Step 4: Preparing the data for stm

Continuing from step 3 in the previous example...

```
out <- stm::readCorpus(dtm, type = 'slam')
```

Step 5: Fitting the model

```r
yelp.out <- stm(documents = out$documents,
                vocab = out$vocab,
                K = 20,
                prevalence = ~ business_categories,
                data = df[-empty_rows, ])
class(yelp.out) #STM
```

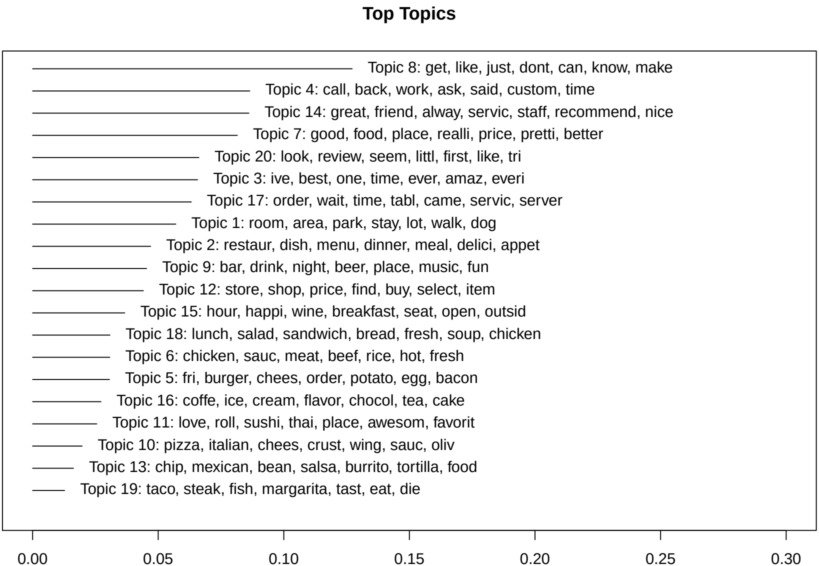
Step 5: Evaluating the topics

```
labelTopics(yelp.out)
```

```
## Topic 1 Top Words:
##       Highest Prob: room, area, park, stay, lot, walk, dog
##       FREX: park, room, stay, hotel, pool, dog, area
##       Lift: central, hotel, park, class, room, stay, train
##       Score: central, park, hotel, room, pool, dog, stay
## Topic 2 Top Words:
##       Highest Prob: restaur, dish, menu, dinner, meal, delici, appet
##       FREX: dish, appet, dinner, restaur, entre, dessert, dine
##       Lift: chile, entre, dish, appet, crab, chop, salmon
##       Score: chile, dish, entre, restaur, dessert, shrimp, appet
## Topic 3 Top Words:
##       Highest Prob: ive, best, one, time, ever, amaz, everi
##       FREX: best, ive, ever, everi, amaz, far, year
##       Lift: mediocr, best, ive, ever, havent, far, valley
##       Score: mediocr, best, ive, ever, year, amaz, phoenix
## Topic 4 Top Words:
##       Highest Prob: call, back, work, ask, said, custom, time
##       FREX: call, told, car, custom, said, work, charg
##       Lift: word, phone, told, answer, call, car, issu
##       Score: word, car, told, call, custom, manag, phone
## Topic 5 Top Words:
##       Highest Prob: fri, burger, chees, order, potato, egg, bacon
##       FREX: burger, fri, egg, bacon, potato, onion, lettuc
##       Lift: burger, lettuc, bun, fri, bacon, egg, onion
##       Score: lettuc, burger, fri, egg, potato, onion, bacon
## Topic 6 Top Words:
##       Highest Prob: chicken, sauc, meat, beef, rice, hot, fresh
```

Step 6: Visualizing the topics

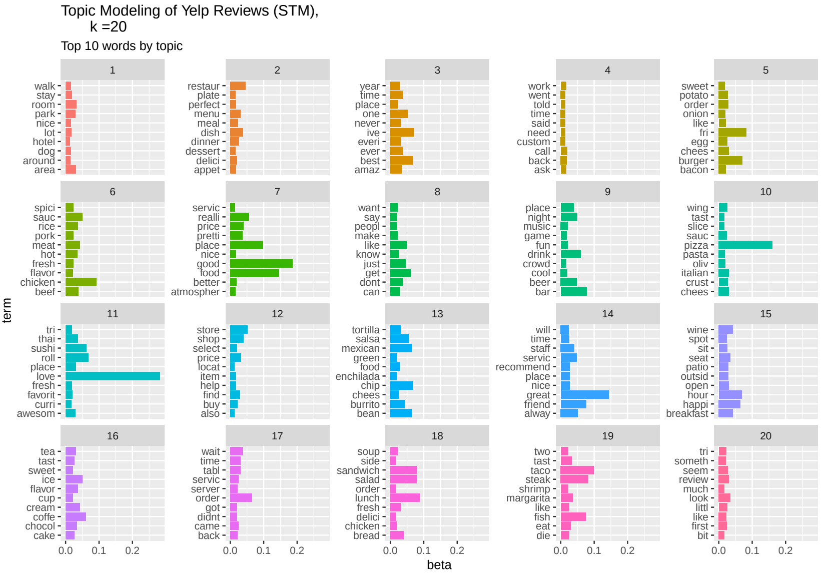
```
plot.STM(yelp.out,type="summary",xlim=c(0,0.3), n=7)
```



Step 6: Visualizing the topics

```
yelp.out %>%
tidy() %>%
group_by(topic) %>%
top_n(10, beta) %>%
ggplot(aes(x = term, y = beta)) +
geom_col() +
coord_flip() +
facet_wrap(~ topic, scales = ’free_y’, nrow = 4) +
labs(title = ’Topic Modeling of Yelp Reviews (STM), k =20’, subt
```

Step 6: Visualizing the topics


Topic Modeling of Yelp Reviews (STM),
k =20
Top 10 words by topic
term
beta
walk
stay
room
park
nice
lot
hotel
dog
around
area
restaur
plate
perfect
menu
meal
dish
dinner
dessert
delici
appet
year
time
place
one
never
ive
everi
ever
best
amaz
work
went
told
time
said
need
custom
call
back
ask
sweet
potato
order
onion
like
fri
egg
chees
burger
bacon
spici
sauc
rice
pork
meat
hot
fresh
flavor
chicken
beef
servic
realli
price
pretti
place
nice
good
food
better
atmospher
want
say
peopl
make
like
know
just
get
dont
can
place
night
music
game
fun
drink
crowd
cool
beer
bar
wing
tast
slice
sauc
pizza
pasta
oliv
italian
crust
chees
tri
thai
sushi
roll
place
love
fresh
favorit
curri
awesom
store
shop
select
price
locat
item
help
find
buy
also
tortilla
salsa
mexican
green
food
enchilada
chip
chees
burrito
bean
will
time
staff
servic
recommend
place
nice
great
friend
alway
wine
spot
sit
seat
patio
outsid
open
hour
happi
breakfast
tea
tast
sweet
ice
flavor
cup
cream
coffe
chocol
cake
wait
time
tabl
servic
server
order
got
didnt
came
back
soup
side
sandwich
salad
order
lunch
fresh
delici
chicken
bread
two
tast
taco
steak
shrimp
margarita
like
fish
eat
die
tri
someth
seem
review
much
look
littl
like
first
bit