Topic 06 - Topic Analysis

Julia Parish

2022-05-10

Topic Analysis

This text sentiment analysis was completed as an assignment for the course, Environmental Data Science 231: Text and Sentiment Analysis for Environmental Problems. The data was sourced from ...

Original assignment instructions can be found here

Load Libraries

```
#install packages as necessary, then load libraries
if (!require(librarian)){
  install.packages("librarian")
  library(librarian)
}

librarian::shelf(here,
                 igraph,
                 kableExtra,
                 ldatuning,
                 LDAvis,
                 LexisNexisTools,
                 lubridate,
                 pdftools,
                 quanteda,
                 quanteda.textplots,
                 quanteda.textstats,
                 readr,
                 reshape2,
                 sentimentr,
                 tidyr,
                 tidytext,
                 tidyverse,
                 tm,
                 topicmodels,
                 tsne)
```

Assignment: run three Topic Analysis models and select the overall best value for k (the number of topics).

Include justification for the selection: theory, FindTopicsNumber() optimization metrics, interpretability, LDAvis

Table 1: EPA Article Statistics

Text	Types	Tokens	Sentences	Document
text1	1196	3973	178	1_Air Alliance.pdf
text2	830	2509	111	10_Bus NEJ.pdf
text3	279	571	31	11_Carlton Ginny.pdf
text4	0	0	0	12_City of Baltimore.pdf
text5	1059	4050	123	13_City of Grandview.pdf
text6	5	5	1	14_City of Phoenix Comment on EJ 2020 Framework.pdf
text7	1745	6904	251	15_City Project.pdf
text8	581	1534	49	16_Corporate EEC.pdf
text9	469	1187	53	17_Detriot Sierra Club.pdf
text10	424	903	38	18_District DOE.pdf
text11	3622	22270	655	19_Earth Justice.pdf
text12	373	717	25	2_Alex Kidd.pdf
text13	404	971	42	20_Elizabeth Mooney.pdf
text14	710	2190	77	21_Env COS.pdf
text15	636	1896	82	22_Env Def Fund.pdf

Load the data

```
comments_df <- read_csv(here("assignments/HW06_TopicAnalysis/data/comments_df.csv"))
```

Create Corpus of EPA Articles

```
epa_corp <- corpus(x = comments_df, text_field = "text")
epa_corp.stats <- summary(epa_corp)

head(epa_corp.stats, n = 15) %>%
  knitr::kable(caption = "EPA Article Statistics") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Tokenize Corpus

```
toks <- tokens(epa_corp, remove_punct = TRUE, remove_numbers = TRUE)

# project-specific stop words here
add_stops <- c(stopwords("en"),"environmental", "justice", "ej", "epa", "public", "comment")

toks1 <- tokens_select(toks, pattern = add_stops, selection = "remove")
```

Convert EPA Corpus Tokens to a Document-Feature Matrix

```
# convert tokens to dfm
dfm_comm <- dfm(toks1, tolower = TRUE)

# stem words in dfm
dfm <- dfm_wordstem(dfm_comm)

# remove terms only appearing in one doc (min_termfreq = 10)
```

```
dfm <- dfm_trim(dfm, min_docfreq = 2)

# remove rows (docs) with all zeros
sel_idx <- slam::row_sums(dfm) > 0

# comments_df <- dfm[sel_idx, ]
dfm <- dfm[sel_idx, ]
```

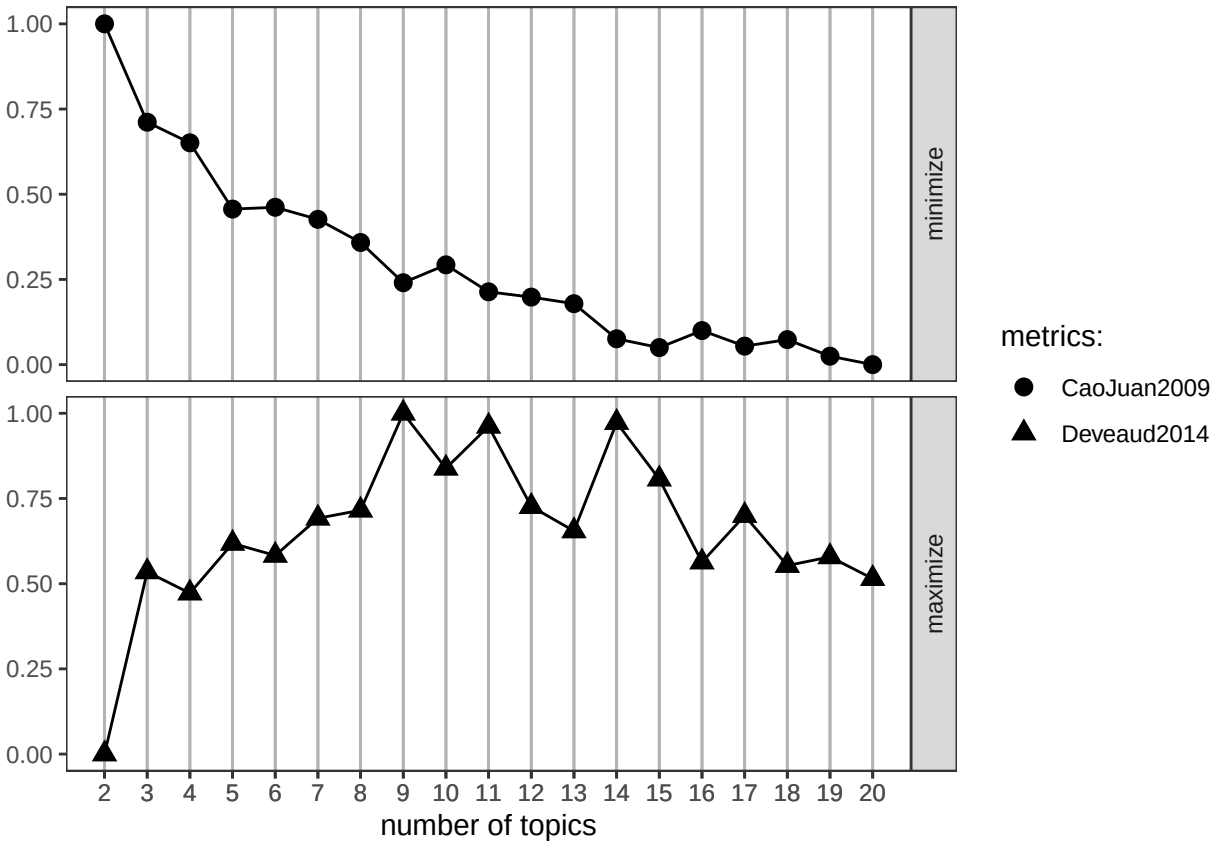
Calculate metrics from the data

CaoJuan 2009 & Deveaud2014 method

```
result_cjD <- FindTopicsNumber(dfm,
                               topics = seq(from = 2, to = 20, by = 1),
                               metrics = c("CaoJuan2009",  "Deveaud2014"),
                               method = "Gibbs",
                               control = list(seed = 77),
                               verbose = TRUE)
```

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

```
FindTopicsNumber_plot(result_cjD)
```



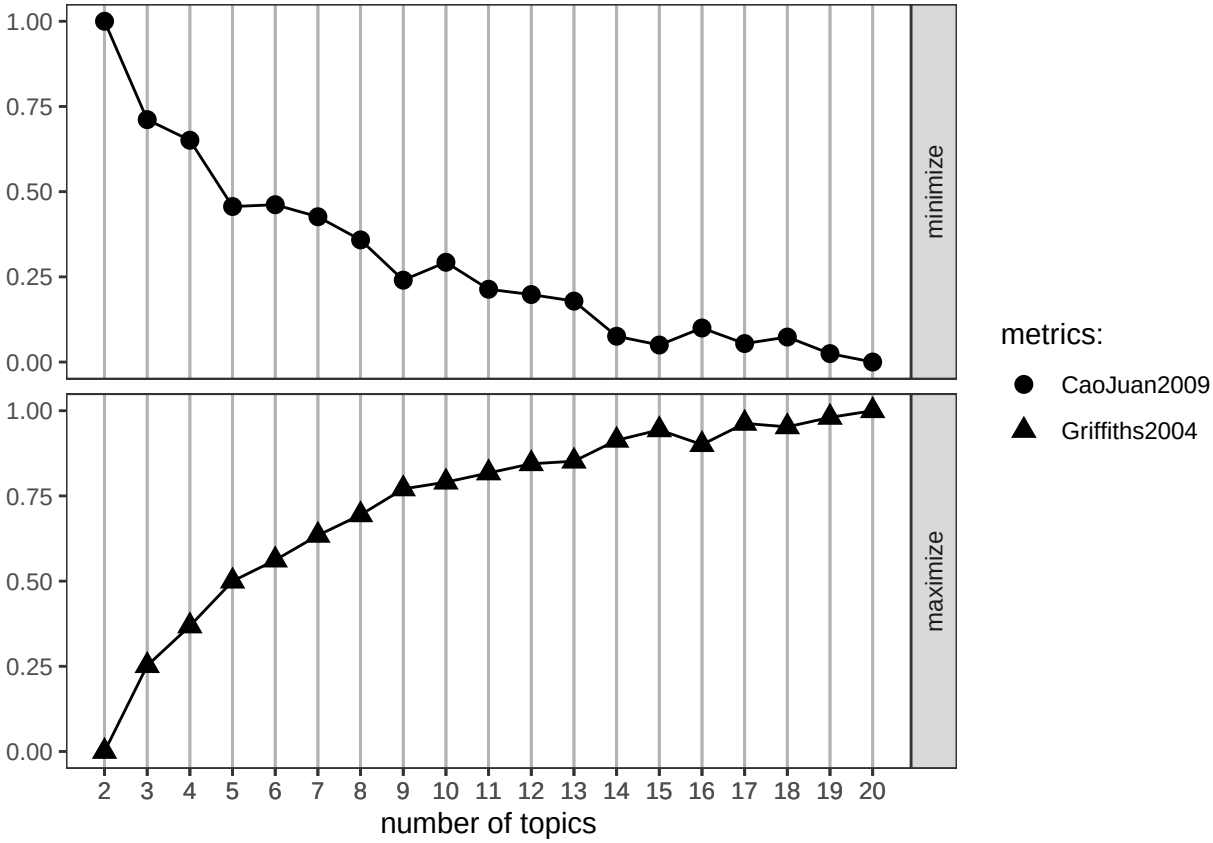
With the CaoJuan 2009 & Deveaud2014 method, it seems like 9 topics is the ideal number.

CaoJuan2009 & Griffiths2004 Method

```
result_cjG <- FindTopicsNumber(dfm,
                               topics = seq(from = 2, to = 20, by = 1),
                               metrics = c("CaoJuan2009", "Griffiths2004"),
                               method = "Gibbs",
                               control = list(seed = 77),
                               verbose = TRUE)
```

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

```
FindTopicsNumber_plot(result_cjG)
```



With the CaoJuan 2009 & Deveaud2014 method, it seems like either 5 or 9 topics could be the ideal number.

Latent Dirichlet Allocation (LDA) Modelling

Model 1: k = 7

Choosing 7 as it lies between 5 topics and 9 topics as a reference

```
# select topic areas and assign to 'k'
k <- 7

# running LDA function, telling it how many topics to look for (9), est. 2 matrices
```

```
topicModel_k7 <- LDA(dfm,
                     k,
                     method="Gibbs",
                     control=list(iter = 500, verbose = 25))
```

```
## K = 7; V = 2893; M = 81
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

Model Results

```
# LDA estimated topics, saved result
tmResult1 <- posterior(topicModel_k7)

# beta matrix from results
beta1 <- tmResult1$terms

#
terms(topicModel_k7, 10)
```

```
##       Topic 1   Topic 2      Topic 3     Topic 4     Topic 5     Topic 6
##  [1,] "agenc"   "communiti"  "communiti" "prison"    "communiti" "state"
##  [2,] "right"   "water"      "peopl"     "permit"    "enforc"    "impact"
##  [3,] "titl"    "pollut"     "citi"      "state"     "permit"    "popul"
##  [4,] "civil"   "health"     "health"    "like"      "comment"   "rule"
##  [5,] "vi"      "new"        "project"   "consid"    "includ"    "health"
##  [6,] "issu"    "reduc"      "can"       "use"       "monitor"   "pollut"
##  [7,] "work"    "overburden" "park"      "grant"     "air"       "air"
##  [8,] "includ"  "clean"      "us"        "carolina"  "complianc" "also"
##  [9,] "address" "comment"    "chang"     "implement" "requir"    "agenc"
## [10,] "feder"   "energi"     "access"    "comment"   "action"    "must"
##       Topic 7
##  [1,] "framework"
```

```
##  [2,] "state"
##  [3,] "draft"
##  [4,] "communiti"
##  [5,] "comment"
##  [6,] "action"
##  [7,] "agenc"
##  [8,] "epa"
##  [9,] "develop"
## [10,] "program"
```

Visualize Model Results

```
svd_tsne <- function(x) tsne(svd(x)$u)

json1 <- createJSON(
  phi = tmResult1$terms,
  theta = tmResult1$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab=""))

serVis(json1)
```

Model 2: k = 5

```
# select topic areas and assign to 'k'
k <- 5

# running LDA function, telling it how many topics to look for (9), est. 2 matrices

topicModel_k5 <- LDA(dfm,
                     k,
                     method="Gibbs",
                     control=list(iter = 500, verbose = 25))
```

```
## K = 5; V = 2893; M = 81
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
```

```
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

Model Results

```
# LDA estimated topics, saved result
tmResult2 <- posterior(topicModel_k5)

# beta matrix from results
beta2 <- tmResult2$terms

#
terms(topicModel_k5, 10)
```

```
##       Topic 1     Topic 2     Topic 3     Topic 4     Topic 5
##  [1,] "state"     "framework" "communiti" "right"     "communiti"
##  [2,] "impact"    "state"     "enforc"    "civil"     "prison"
##  [3,] "pollut"    "draft"     "comment"   "plan"      "water"
##  [4,] "communiti" "communiti" "includ"    "communiti" "work"
##  [5,] "health"    "agenc"     "action"    "health"    "local"
##  [6,] "air"       "program"   "air"       "vi"        "citi"
##  [7,] "provid"    "develop"   "monitor"   "includ"    "agenda"
##  [8,] "must"      "comment"   "region"    "titl"      "comment"
##  [9,] "rule"      "effort"    "permit"    "agenc"     "make"
## [10,] "guidanc"   "use"       "complianc" "peopl"     "year"
```

Visualize Model Results

```
svd_tsne <- function(x) tsne(svd(x)$u)

json2 <- createJSON(
  phi = tmResult2$terms,
  theta = tmResult2$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab=""))

serVis(json2)
```

Model 3: k = 9

```
# select topic areas and assign to 'k'
k <- 9

# running LDA function, telling it how many topics to look for (9), est. 2 matrices
```

```
topicModel_k9 <- LDA(dfm,
                     k,
                     method="Gibbs",
                     control=list(iter = 500, verbose = 25))
```

```
## K = 9; V = 2893; M = 81
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

Model Results

```
# LDA estimated topics, saved result
tmResult3 <- posterior(topicModel_k9)

# beta matrix from results
beta3 <- tmResult3$terms

#
terms(topicModel_k9, 10)
```

```
##       Topic 1   Topic 2     Topic 3     Topic 4     Topic 5    Topic 6
##  [1,] "agenc"   "communiti" "communiti" "framework" "state"    "communiti"
##  [2,] "program" "impact"    "pollut"    "communiti" "permit"   "plan"
##  [3,] "state"   "health"    "air"       "draft"     "consid"   "local"
##  [4,] "feder"   "state"     "health"    "action"    "use"      "govern"
##  [5,] "issu"    "rule"      "comment"   "effort"    "comment"  "strategi"
##  [6,] "titl"    "pollut"    "reduc"     "agenda"    "opportun" "use"
##  [7,] "act"     "also"      "protect"   "comment"   "organ"    "help"
##  [8,] "right"   "air"       "develop"   "epa"       "feder"    "action"
##  [9,] "polici"  "ejscreen"  "polici"    "develop"   "air"      "need"
## [10,] "vi"      "asthma"    "p"         "water"     "process"  "particip"
##       Topic 7   Topic 8     Topic 9
##  [1,] "prison"  "communiti" "health"
##  [2,] "facil"   "enforc"    "communiti"
```

```
##  [3,] "energi"  "comment"   "citi"
##  [4,] "project" "includ"    "peopl"
##  [5,] "water"   "monitor"   "includ"
##  [6,] "popul"   "provid"    "park"
##  [7,] "site"    "use"       "can"
##  [8,] "sourc"   "report"    "law"
##  [9,] "center"  "action"    "see"
## [10,] "peopl"   "region"    "project"
```

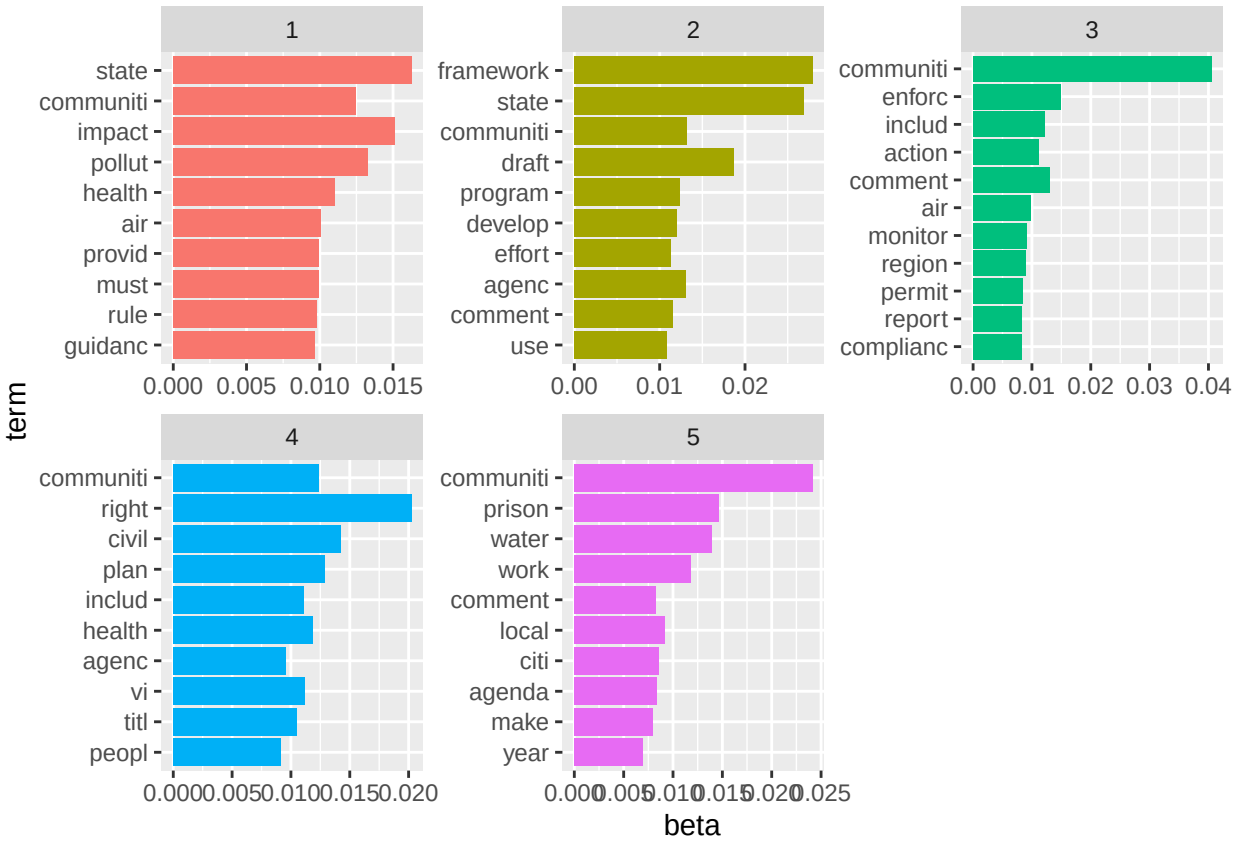
Visualize Model Results

```
svd_tsne <- function(x) tsne(svd(x)$u)

json3 <- createJSON(
  phi = tmResult3$terms,
  theta = tmResult3$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab=""))

serVis(json3)
```

Top Topic Terms for Model 2: k = 5

```
comment_topics <- tidy(topicModel_k5, matrix = "beta")

top_terms <- comment_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

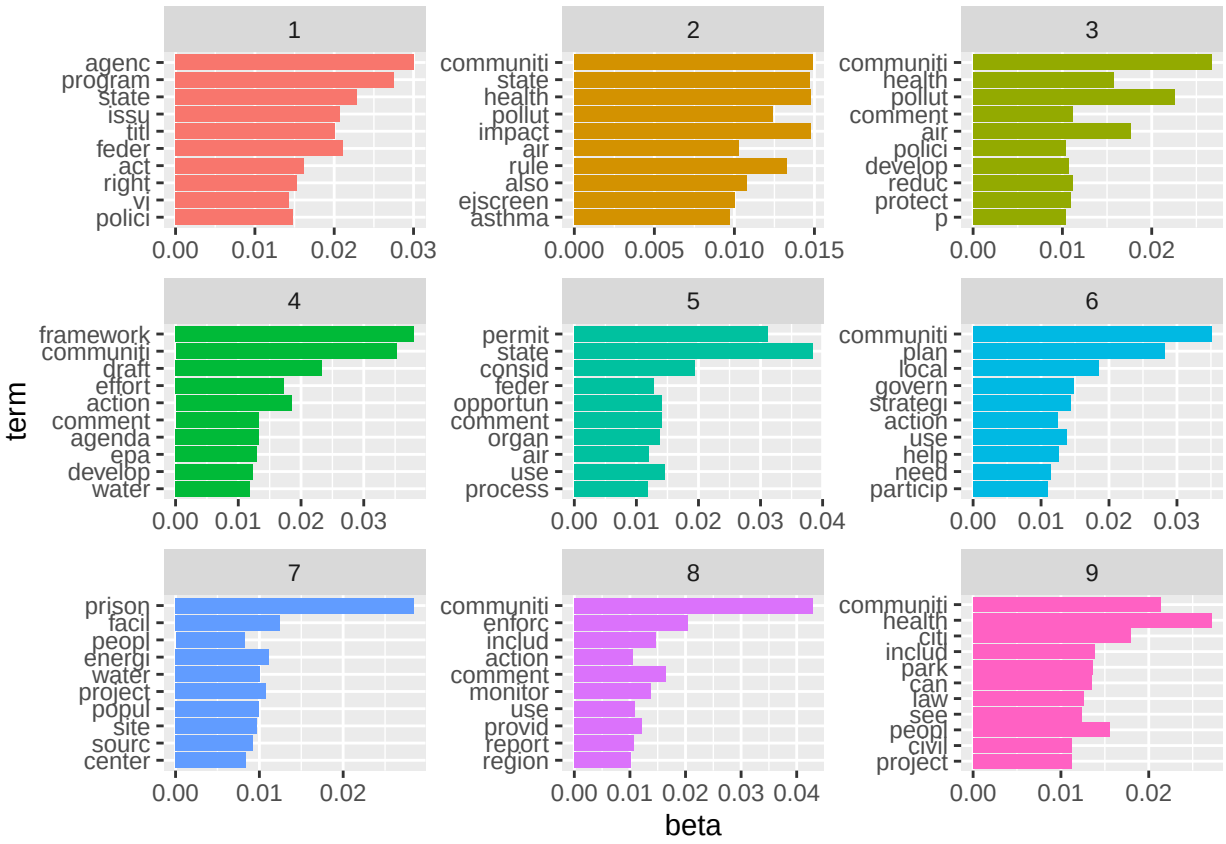
```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



Top Topic Terms for Model 3: k = 9

```
comment_topics9 <- tidy(topicModel_k9, matrix = "beta")

top_terms9 <- comment_topics9 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```
top_terms9 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



Result Response

It seems like 5 topics is the best selection due to the distance between each topic groups in the `json servis` visualization. When plotting the top topic terms, 5 topics also seems like clearer divisions than 9 topics.