Lab #7: Structural Topic Modeling

Due in Sakai March 22, 2019 (by 10:00a)

Introduction

For this lab, you will generate and interpret your own structural topic model (STM) analysis.

Getting Started

First, set your working directory. Open up a fresh RStudio session (if you open RStudio and a previous work session is loaded, be sure you save any R and/or RData files before you close them out). Then set your working directory:

```
setwd("working_directory_here")
```

Then open up a fresh R script. Save it using your first initial, full last name, and then "_lab7." So my script would be titled **MTaylor_lab7.R**.

You should always strive to keep your scripts tidy. At the top of your R script, type this (substituting in your first initial and last name):

```
##############################
##  MTaylor_lab7.R
##  Note: Code for lab assignment #7
##  Author: Marshall A. Taylor
##############################

###BEGIN###
```

You are ready to begin your code. Be sure to include all the code necessary for me to check your work. Document your code thoroughly (using "#").

Remember that saving your R work is a two-step process. You save your R script using Cmd+Enter (Mac) or Cntrl+Enter (Windows). You save your R data objects like this:

```
save.image("MTaylor_lab7.RData")
```

Lastly, prep a Word, Pages, or LaTeX document that has the same title structure: e.g., **MTaylor_lab7.docx**. This is where you put your write-ups and visualizations.

You should turn in three documents to Sakai: your **R script** that shows the code you used, **RData file** that provides the data your scraped, and **Word document** (or whatever text processor you choose to use) with your write-ups.

Assignment

For this lab, we are going to use the UN General Debate Corpus. The texts were compiled by Alexander Baturo, Niheer Dasandi, and Slava Mikhaylov (2017). During the annual United Nations General Assembly, a representative from each UN member state provides a statement "that present[s] their government's perspective on the major issues of world politics. These statements are akin to the annual legislative state-of-the-union addresses in domestic politics"" (Mikhaylov, Baturo, and Dasandi 2017).

The corpus contains the statements from 1970 to 2017—7,897 speeches total. However, you are going to focus on the 2017 statements made available through the `quanteda.corpora` package (Benoit, Watanabe, and M"uller 2019), which contains 196 statements. I have removed those statements that contain missing data on the two covariates I have you use, which brings your sample down to 187 documents.

You are going to focus on how topic prevalence/content varies as a function of two variables: the continent on which the country resides ("continent") and the country's GDP (gross domestic product) per capita for 2017 ("gdp_per_capita").

The provided corpus is labeled "un_corpus.rds." It can be read in using the `readRDS()` function.

Preprocess the corpus in whatever way you think is appropriate. **There is one extra thing I want you to do in the preprocessing stage that we didn't do in class: Somhere in the `prepDocuments()` function, include `upper.thresh = 170`. This will remove very common words—specifically, those that show up in at least 170 of the 187 statements. I am having you do this because, like the SOTUs, there's a lot of repeating words like "nation," "country," and so on.**

Once you have cleaned up the texts, do the following:

1. Using semantic coherence and exclusivity metrics, determine the number of topics (k) you should use. Create a semantic coherence-by-exclusivity scatterplot and paste it into your Word document. According to these statistics, how many topics should you go with? Why is it important to consider semantic coherence *in conjunction* with exclusivity when determining how many topics to use?

2. Pick the k that strikes the "best balance," in your view, between semantic coherence and exclusivity. Now estimate some topics using the `stm()` function, where topic prevalence is conditioned on the "continent"" and "gdp_per_capita" variables. Be sure to use the b-spline transformation on the GDP variable (i.e., `s(gdp_per_capita)`). You only have 187 documents, so take out the `max.em.its` bit.

3. Create a summary plot that shows (1) the topics in descending order of prominence in the corpus, and (2) the 5 highest probabiity terms associated with each topic. Create another summary plot that again plots the topics in decending order of prominence in the corpus, but this time shows the 5 terms per topic with the highest lift. Copy and paste these plots into your Word document.

4. Create an interactive visualization using the `stmBrowser` package. When you submit this, you will want to upload the folder that the `stmBrowser` package outputs. This folder should be called "stm-visualization." You will want to submit that entire folder (not just the files contained therein). You may have to compress the folder in order to submit it via Sakai.

5. Using the visuals you have created so far, interpret 3 of the topics. You may find the statement texts outputted by `stmBrowser` and the `findThoughts()` function particularly useful for finding representative statements for each topic. Give each of these topics a label (like how we called the fossil fuels topic in the political blogs corpus the "Fossil Fuels" topic).

6. Estimate the effects of the "continent" and "gdp_per_capita" variables on these 3 topics.

7. Plot the prevalence differences in these 3 topics between two continents of your choosing. Copy and paste the plot into your Word document. What does the plot tell you? (FYI: There are five continents in the dataset: Africa, Americas, Asia, Europe, and Oceania. Also, you will probably need to extend your x-axis range to accomodate the length of the confidence intervals.)

8. Does engagement in any of these 3 topics vary as a function of the delegate's home country's GDP per capita? Create a plot showing this. Copy and paste the plot into your Word document. What does the plot tell you? Are there any meaningful associations between topic prevalence and GDP per capita? (FYI: If you are referencing the topic-over-time graph we made in class in order to make this plot, you will want to remove the `xaxt = "n"` bit, and remove everything between the `plot()` function and the `legend()` function since we aren't trying to convert dates. Also, you only have 187 documents, so don't be surprised if your confidence intervals are really wide.)

9. **Extra Credit:** Are there any meaningful differences in how any of these 3 topics are discussed (i.e., differences in topic content)? Create some plots showing these content differences. Since you are dealing with a categorical variable ("continent") with more than two levels, you will want to specify which levels you are interesting in contrasting within the `plot()` function with this: `covarlevels = c("continent1", "content2")`, where "continent1" and "continent2" are the two continents you want to contrast.

Hints

The R scripts for the structural topic modeling slides (pres81.R) might be very helpful. Like, *really really helpful*.

This is the preprocessing workflow I used to clean the statements Feel free to customize it any way you want.

```
un.texts <- VCorpus(VectorSource(un_corpus$texts))

#Clean it up
removeAllPunct <- function(x) gsub("[[:punct:]]", " ", x)
removeSpecialChars <- function(x) gsub("[^a-zA-Z0-9 ]", " ", x)

un.texts <- tm_map(un.texts, content_transformer(removeAllPunct))
un.texts <- tm_map(un.texts, content_transformer(removeSpecialChars))
un.texts <- tm_map(un.texts, content_transformer(tolower))
un.texts <- tm_map(un.texts, removeNumbers)
un.texts <- tm_map(un.texts, removeWords, stopwords("english"))
un.texts <- tm_map(un.texts, stripWhitespace)

un.texts <- convert.tm.to.character(un.texts)
un.texts <- lemmatize_strings(un.texts,
                              dictionary = lexicon::hash_lemmas)

un.texts <- VCorpus(VectorSource(un.texts))

inspect(un.texts[[1]])


#Remove some sparse terms
un.texts <- DocumentTermMatrix(un.texts)
un.texts <- removeSparseTerms(un.texts, .99)
```

The the `removeSpecial` bit of code is adapted from here, and the `removeAllPunct` code is adapted from here.

References

Benoit, Kenneth, Kohei Watanabe, and Stefan M"uller. 2019. "`quanteda.corpora`: A Collection of Corpora for `quanteda`." R package version 0.87. Retrieved March 6, 2019 (link).

Baturo, Dasandi, and Mikhaylov. 2017. "Understanding State Preferences with Text as Data: Introducing the UN General Debate Corpus." *Research and Politics* 4(2):1-9.

Mikhaylov, Slava, Alexander Baturo, and Niheer Dasandi. 2017. "United Nations General Debate Corpus." Harvard Dataverse, version 4. Retrieved March 6, 2019 (link).