Structural Topic Modeling

Marshall A. Taylor, Ph.D.

2020 HSI Learning Resilience Conference
4:15p to 5p (EST)
November 3, 2020

Introduction

Recap on topic modeling:

- In the workshop we talked a bit about topic modeling, particularly in the form of latent Dirichlet allocation (LDA).
- The goal of LDA is to identify some representation of the latent thematic structure in a corpus.
- We find the sets of words that tend to co-occur with one another. We call these "topics."

Introduction

Recap on topic modeling, continued:

- Finding this latent thematic structure is conceptualized as a matrix factorization problem. Specifically, we try to decompose the *observed* word-document relationship into *unobserved* document-topic and topic-word relationships (Fligstein, Brundage, and Schultz 2017:888).
- These two unobserved matrices are represented as probability distributions: the probabilistic distribution of documents across topics, and the probabilistic distribution of topics across words.
- We use the topic-word probability matrix to interpret the topics—perhaps by looking at terms with the highest probability and lift scores per topic.
- We use the document-topic probability matrix to examine how prevalent particular topics are within a document or across the corpus as a whole.

Topic Modeling Output – Document-by-Topic

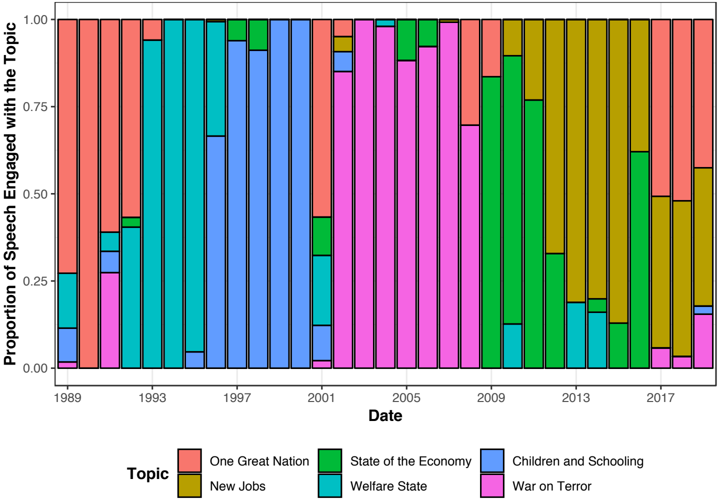


Figure 1: SOTU Topic Engagement Over Time

Extending the Method's Utility

As social scientists, we often want to go further than simply identifying the topics in a corpus.

We may want to ask questions like this:

- *Are advocacy organizations more likely to engage a particular topic if they receive corporate funding (Farrell 2016)?*
- *Do the types of issues detailed in aviation incidence reports differ depending on whether the flight is commercial or private (Kuhn 2018)?*
- *Are employees at companies with poor organizational culture rankings more likely to discuss employee treatment in their workplace reviews than employees at companies with high organizational culture rankings (Schmiedel, Müller, and vom Brocke 2018)?*

Extending the Method's Utility

Another way to put this is that we are often interested in understanding how **topic prevalence** and/or **topic content** varies as a function of other variables.

Topic prevalence refers to the extent to which a topic appears in the corpus or for certain groups of documents in the corpus.

Topic content refers to the particular words that are used to communicate a topic.

Introducing STM

This is where **structural topic modeling (STM)** comes in (Roberts et al. 2013).

We can go a step further beyond simply estimating the latent topic structure. With STM, we can see how other variables or "metadata"—e.g., the author's gender, race-ethnicity, political affiliation, time of publication, and so on—impact topic prevalence and content.

A Regression Framework

STM accomplishes these tasks by embedding topic modeling within a **regression analysis** framework.

Regression is easily the most common tool for quantitative analysis in the social sciences. The most basic type—linear regression, or ordinary least squares (OLS) regression—takes the following form:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \epsilon_i \tag{1}$$

Where Y is the *dependent variable*, and Y_i is the i^{th} case's value on Y. X_{1i} is that case's value on the *independent variable* X_1, and β_1 is the *coefficient* that shows how Y varies as a function of X. The first term on the right-hand side, β_0 is known as the *y*-intercept (or constant), and ϵ_i is called the *residual*, or the unexplained variation in Y not accounted for by X.

A Regression Framework

This isn't the time or place to get into regression modeling in any sort of meaningful way, but the take home point is this:

β_1 is interpreted as how much Y changes with a one-unit change in X.

$$Y_i = \beta_0 + \beta_1 X_{1i} + \epsilon_i \tag{2}$$

So, if Y is, say, weight (in kilograms), X is height (in centimeters), and β_1 is .76, then we would say that for every centimeter taller a person is, we can expect their weight to be higher by .76 kilograms.[1]

[1] This estimate comes from the publicly available sample of the NHANES data available through StataCorp.

A Regression Framework

Once we have estimates of β_0 and β_1, we can also make predictions:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} \tag{3}$$

Where the little hat indicates a *predicted value* (as opposed to an observed value).

So if $\hat{\beta}_0 = -55.39$ and $\hat{\beta}_1 = .76$, then for a person who is 167.65 cm. tall:

$$\hat{Y}_i = -55.39 + (.76 \times 167.65) \tag{4}$$

Based on our model, we expect this hypothetical person to weigh approximately 72.02 kg—or about 158.78 lbs.

A Regression Framework

It's easy enough to have more than one X:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \cdots + \beta_k X_{ki} + \epsilon_i \tag{5}$$

Where X_2 and X_3 could be, say, in the context of our weight example, age and blood pressure. You get the idea.

Topics and Regression

If you imagine Y as one of your topics from a topic model analysis and all of your Xs as some other variables associated with the documents, you have STM. In theory, anyway; the regression equation underlying STM is a bit more complicated than that.

If you want to dig into the regression model underlying STM, take a look at Roberts, Stewart, and Tingley (forthcoming, specifically p. 3).

STM Example

Let's work through an example using the `stm` package (Roberts, Stewart, and Tingley 2015).

Let's use the full dataset from the *CMU Political Blogs Corpus* (Eisenstein and Xing 2010). As a reminder, this is a set of blog posts from across six political blogs collected over the course of 2008. Three of the sources are traditionally identified as liberally-oriented (*Digby*, *ThinkProgress*, and *Talking Points Memo*) and the other three are traditionally identified as conservatively-oriented (*American Thinker*, *Hot Air*, and *Michelle Malkin*).

The dataset contains two useful metadata variables: political affiliation (liberal vs. conservative) and date of publication.

This is the same dataset the `stm` package authors use to detail their package. As such, a lot today's code comes from/is influenced by Roberts et al. (forthcoming; link here).

As always, we'll begin by installing and loading the required packages.

```
install.packages(c("tm", "stm", "ggplot2",
                   "reshape2", "tidytext", "dplyr",
                   "textstem", "textreg", "grid",
                   "devtools", "LDAvis", "servr"))
library(tm)
library(stm)
library(ggplot2)
library(reshape2)
library(tidytext)
library(dplyr)
library(textstem)
library(textreg)
library(grid)
library(LDAvis)
library(servr)
devtools::install_github("mroberts/stmBrowser",
    dependencies = T)
library(stmBrowser)
```

Load in the corpus. It consists of 13,246 blog posts.

```
#Load in data
blogs <- read.csv(file = "poliblogs2008.csv",
    header = T, row.names = 1)
```

Let's take care of some preprocessing.

```r
blog.texts <- VCorpus(VectorSource(blogs$documents))

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

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

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

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

inspect(blog.texts[[1]])

#Convert to a DTM and remove sparse terms
blog.texts <- DocumentTermMatrix(blog.texts)
blog.texts <- removeSparseTerms(blog.texts, .99)
```

Additionally, the stm package requires that the document indices, vocabulary, and metadata be stored in separate objects.

Luckily, all of these steps are handled with the prepDocuments() function:

```
prepped.blogs <- readCorpus(blog.texts, type = "slam")
prepped.blogs <- prepDocuments(
    prepped.blogs$documents, prepped.blogs$vocab,
    blogs[c("rating", "day")])
docs <- prepped.blogs$documents
vocab <- prepped.blogs$vocab
meta <- prepped.blogs$meta
```

Now we're ready to start estimating some topics, which will take the following syntactical form:

```
model <- stm(documents = docs, vocab = vocab, k = 10,
    init.type = "LDA", prevalence =~ rating + s(day),
    max.em.its = 75, data = meta, verbose = F,
    seed = 123)
```

First, however, let's address something. If you recall, a pressing concern in topic modeling is the selection of *k*—the number of topics. Last week we talked about the most important method for choosing the appropriate number of topics: your brain.

That said, the `stm` package provides some statistics to help guide your decision-making process.

The package allows us to run a number of different topic model solutions with different values for *k* and then compare diagnostic test statistics across them. We do this with the `searchK()` function:

```
k.number <- searchK(documents = docs, vocab = vocab,
    K = seq(10, 50, by = 10), data = meta,
    prevalence =~ rating + s(day), proportion = 0.5,
    heldout.seed = 123, seed = 123, init.type = "LDA",
    verbose = T)
plot(k.number)
```

This chunk of code is in the class script, but don't bother trying to run it. It takes a long time to run.

This also means that you should plan accordingly when you use the `searchK()` function on your own.

searchK() Output

Here's the output from the `searchK()` function:

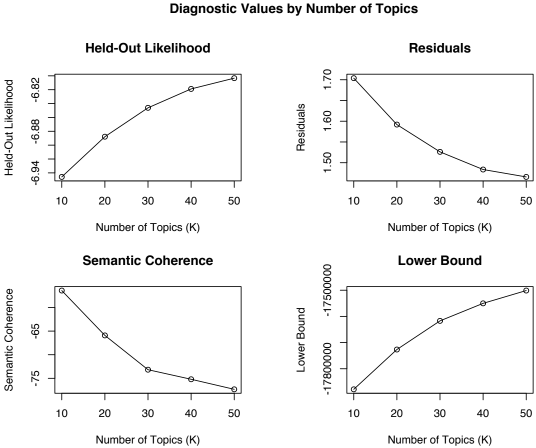


Figure 2: Topic Model Diagnostics, from $k = 10$ to $k = 50$

How Many Topics?

There's a lot of information in this plot. What do we make of it?

The plot shows four diagnostics that help guide our selection of k: semantic coherence (Minmo et al. 2011), document-completion heldout log-likelihood (Wallabach et al. 2009), residual overdispersion (Taddy 2012), and the lower bound.

The statistic I want to focus on for the sake of time and interpretability is **semantic coherence**.

Semantic Coherence

The semantic coherence for the k^{th} topic is found with (Roberts et al. forthcoming:12):

$$C_k = \sum_{i=2}^{M} \sum_{j=1}^{i-1} \log \left(\frac{D(v_i, v_j) + 1}{D(v_j)} \right) \tag{6}$$

Where $D(v_j)$ is the number of documents where word v_j appears at least once, $D(v_i, v_j)$ is the number of documents where words v_i and v_j occur together, and M is a list of the most probable words associated with topic k.

Semantic coherence should always be negative. Values closer to 0 indicate a topic with good semantic coherence; more negative values indicate a topic with bad semantic coherence.

Semantic Coherence

We are usually more interested in the corpus-level semantic coherence for a topic model solution:

$$C_{K|K=\{k_1,k_2,\cdots,k_k\}} = \frac{C_{k_1} + C_{k_2} + \cdots + C_{k_k}}{K} \tag{7}$$

This is just the mean semantic coherence for a given topic model solution. This is what is reported in the `searchK()` plot.

Semantic Coherence

The semantic coherence statistic quantifies the extent to which high-probability terms in the topics co-occur together. Since topics are supposed to be composed of terms that co-occur together, topics with high-probability terms that rarely co-occur with one another might signal a junk topic.

If this is the case across topics, the semantic coherence value will be smaller (i.e., more negative, larger absolute value), and that *k* may not be the best.

How Many Topics?

Look back at the diagnostics plot. If we were basing our decision purely on semantic coherence, how many topics would we choose?

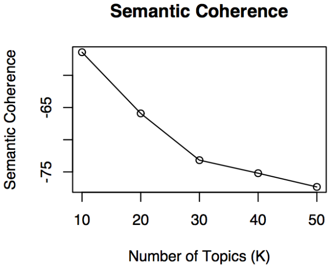


Figure 3: Average Semantic Coherence, from $k = 10$ to $k = 50$

How Many Topics?

I'd go with 10 topics. One issue, though, is that semantic coherence tends to be negatively correlated with the number of topics. Can you think of why that might be problematic?

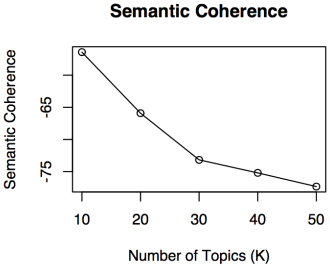


Figure 4: Average Semantic Coherence, from $k = 10$ to $k = 50$

Exclusivity

So semantic coherence will tend to (but not always) favor models with smaller k.

It is therefore wise to use semantic coherence in combination with another diagnostic known as **exclusivity**.

Exclusivity is a measure of the extent to which a particular topic tends to monopolize the presence of particular words. Here's how we define it formally (Bischof and Airoldi 2012:3):

$$E_{wk} = \frac{f_{wk}}{\sum_{j \in K | j \neq k} f_{wj}} \tag{8}$$

So exclusivity is the frequency of word w in topic k relative to the sum of the frequencies of word w in all of the other topics. Larger values indicate the word is more exclusive to that topic.

Exclusivity

We can take the average exclusivity across all topics just like we did with semantic coherence.

Unlike semantic coherence, exclusivity is often *positively* with the number of topics.

So our decision on k should balance semantic coherence with exclusivity.

Here's how we'll do it:

```
ggplot(data = k.number$results, aes(x = semcoh,
    y = exclus)) +
  geom_text(aes(label = K)) +
  xlab("Semantic␣Coherence") +
  ylab("Exclusivity") +
  theme_bw() +
  theme(axis.title = element_text(face = "bold"),
        panel.grid = element_blank())
```

How Many Topics?

What *k* should we go with, if we base our decision strictly on these statistics?

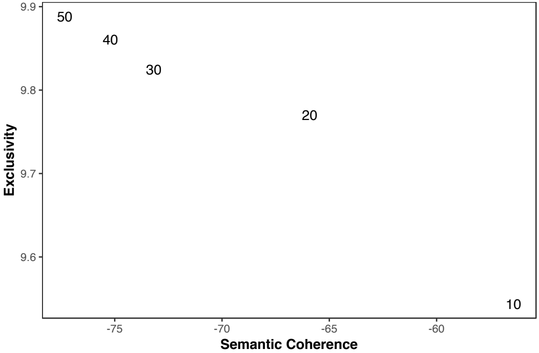


Figure 5: Average Semantic Coherence by Average Exclusivity, from $k = 10$ to $k = 50$

Now we're ready to estimate the topics. Let's go with 20:

```
#Estimate some topics
blog.topics <- stm(documents = docs, vocab = vocab,
    K = 20, data = meta, prevalence =~ rating + s(day),
    seed = 123, init.type = "LDA", max.em.its = 75,
    verbose = T)
```

This takes a while to run. So for now, just load in a pre-estimated model:

```
blog.topics <- readRDS(file = "blog_topics1.rds")
```

Top Terms

What are the highest probability terms per topic?

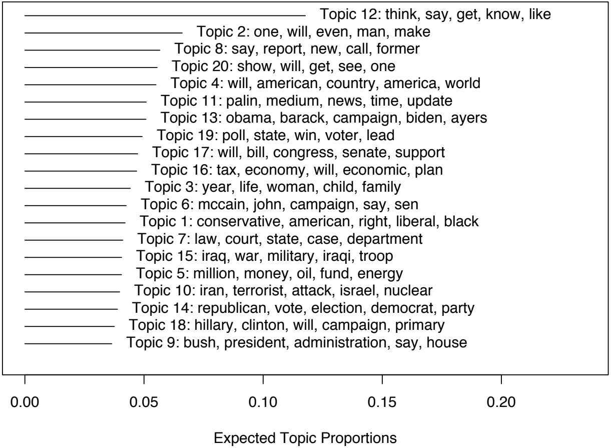


Figure 6: Highest Probability Terms per Topic

Top Terms

What are the highest lift terms per topic?

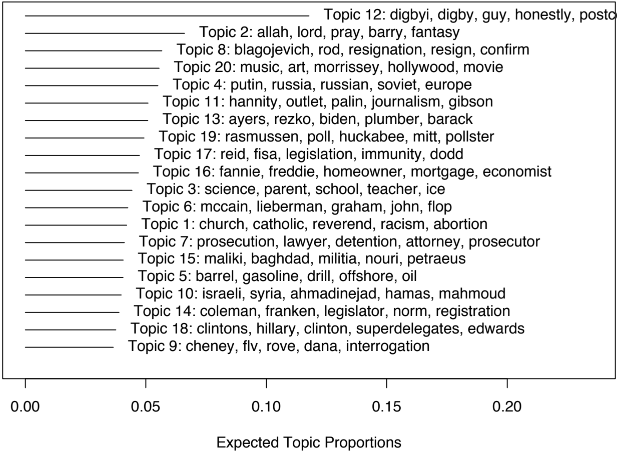


Figure 7: Highest Lift Terms per Topic

Another way to look at the top terms:

```
labelTopics(blog.topics)
```

Can you interpret any of these topics based probability and lift?

The stm package also includes a cool function that let's you see documents that load highly on the topics.

Consider topic #18, which is clearly a “Hillary Clinton Presidential Campaign” topic. What's the text for the three blogs with the highest probability of engaging this topic?

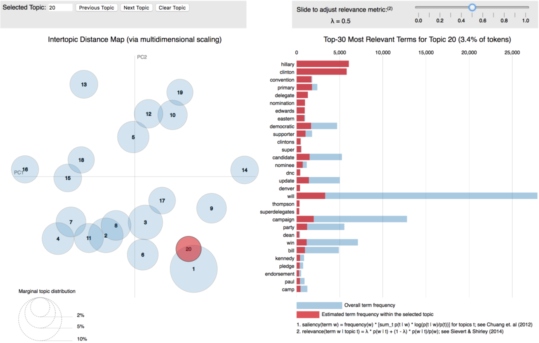
```
#Top documents associated with topic #18
findThoughts(blog.topics, texts =
    as.character(blogs$documents), n = 3, topics = 18)
```

We can also use LDAvis like we did with the topicmodels package to compare term relevance at different λ weighting factors, though it requires a little bit of extra work:

```r
logbeta <- blog.topics$beta$logbeta
margbeta <- exp(logbeta[[1]])
if(length(logbeta) > 1) {
  weights <- blog.topics$settings$covariates$betaindex
  tab <- table(weights)
  weights <- tab/sum(tab)
  margbeta <- margbeta*weights[1]
  for(i in 2:length(blog.topics$beta$logbeta)) {
    margbeta <- margbeta + exp(blog.topics$beta$logbeta[[i]])*weights[i]
  }
}

json <- createJSON(phi = margbeta,
                   theta = blog.topics$theta,
                   doc.length = rowSums(as.matrix(blog.texts)),
                   term.frequency = colSums(as.matrix(blog.texts)),
                   vocab = vocab)
serVis(json, out.dir = "vis", open.browser = interactive())
```

Interactive Visualization



Selected Topic: 20

Slide to adjust relevance metric:(2) λ = 0.5

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

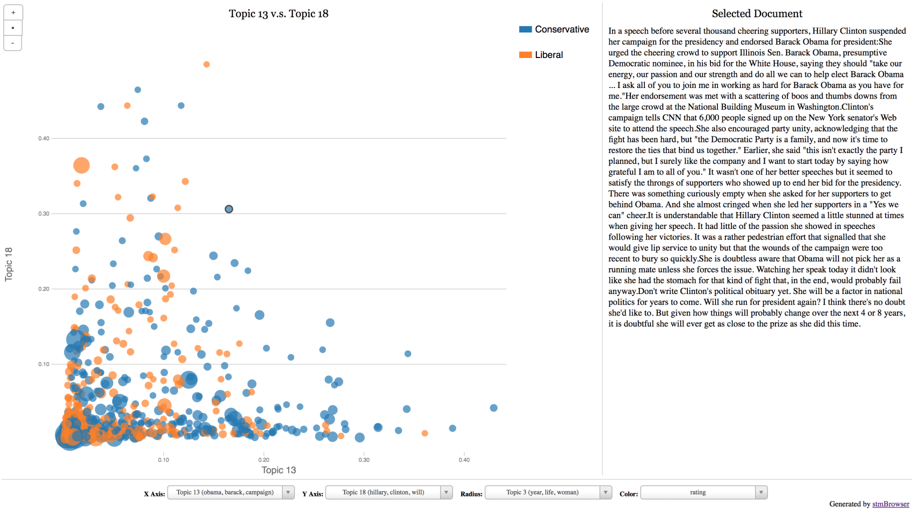
Figure 8: Screenshot from `LDAvis` Output

Remember: The topic numbers between your estimated model and `LDAvis` probably won't align. Go by the words, not the topic number.

The authors of the stm package also have their own interactive visualization package: stmBrowser (Freeman et al. 2015). This visualization incorporates our document metadata to see how topic prevalence varies as a function of ideological affiliation and date of publication:

```
stmBrowser(blog.topics, data = blogs,
    c("rating", "day"),
    text = "documents")
        #Note that this is sampling only 1,000
          #blogs, so results may differ
```

Another Interactive Visualization



Selected Document

In a speech before several thousand cheering supporters, Hillary Clinton suspended her campaign for the presidency and endorsed Barack Obama for president:She urged the cheering crowd to support Illinois Sen. Barack Obama, presumptive Democratic nominee, in his bid for the White House, saying they should "take our energy, our passion and our strength and do all we can to help elect Barack Obama ... I ask all of you to join me in working as hard for Barack Obama as you have for me."Her endorsement was met with a scattering of boos and thumbs downs from the large crowd at the National Building Museum in Washington.Clinton's campaign tells CNN that 6,000 people signed up on the New York senator's Web site to attend the speech.She also encouraged party unity, acknowledging that the fight has been hard, but "the Democratic Party is a family, and now it's time to restore the ties that bind us together." Earlier, she said "this isn't exactly the party I planned, but I surely like the company and I want to start today by saying how grateful I am to all of you." It wasn't one of her better speeches but it seemed to satisfy the throngs of supporters who showed up to end her bid for the presidency. There was something curiously empty when she asked for her supporters to get behind Obama. And she almost cringed when she led her supporters in a "Yes we can" cheer.It is understandable that Hillary Clinton seemed a little stunned at times when giving her speech. It had little of the passion she showed in speeches following her victories. It was a rather pedestrian effort that signalled that she would give lip service to unity but that the wounds of the campaign were too recent to bury so quickly.She is doubtless aware that Obama will not pick her as a running mate unless she forces the issue. Watching her speak today it didn't look like she had the stomach for that kind of fight that, in the end, would probably fail anyway.Don't write Clinton's political obituary yet. She will be a factor in national politics for years to come. Will she run for president again? I think there's no doubt she'd like to. But given how things will probably change over the next 4 or 8 years, it is doubtful she will ever get as close to the prize as she did this time.

X Axis: Topic 13 (obama, barack, campaign) **Y Axis:** Topic 18 (hillary, clinton, will) **Radius:** Topic 3 (year, life, woman) **Color:** rating

Generated by stmBrowser

Figure 9: Screenshot from `stmBrowser` Output

Topic Prevalence

Let's say we've interpreted and labeled all 20 topics. We've decided that topic #18 is a "Hillary Clinton Presidential Campaign" topic #15 is an "Iraq War" topic, and topic #5 is a "Fossil Fuels" topic.

Now we want to answer the following questions:

1. Are liberal bloggers or conservative bloggers more likely to talk about these topics?
2. Did engagement in these topics vary over time?

This is where the regression stuff comes in. We want to see how the prevalence of these topics varies as a function of our document metadata.

First, we need to estimate our regression equations:

```
#Topic prevalence
blog.effects <- estimateEffect(formula = c(5, 15, 18)
    ~ rating + s(day),stmobj = blog.topics,
    metadata = meta)
summary(blog.effects)
```

Topic Prevalence

Table 1: Topic Prevalence by Political Ideology

	Fossil Fuels	**Iraq War**	**Clinton Campaign**
Liberal	-0.005***	0.007**	-0.001
	(0.002)	(0.002)	(0.002)
Constant	0.014*	0.050***	0.083***
	(0.007)	(0.002)	(0.009)
N	13,236	13,246	13,246

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

This is how we'd traditionally present regression results. Does anyone want to take a shot at interpreting the coefficients for "Liberal"?

Topic Prevalence

Table 2: Topic Prevalence by Political Ideology

	Fossil Fuels	**Iraq War**	**Clinton Campaign**
Liberal	-0.005***	0.007**	-0.001
	(0.002)	(0.002)	(0.002)
Constant	0.014*	0.050***	0.083***
	(0.007)	(0.002)	(0.009)
N	13,236	13,246	13,246

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

This is not a statistics class, so I do not expect you to report tables like this. Don't worry if you've never seen a table like this before.

Instead of looking at coefficients, it's probably easier to look at some actual predicted values. In other words, what are the predicted topic proportions for liberal and conservative blog posts across these three topics?

```
#Plot the prevalence differences
plot(blog.effects, covariate = "rating",
    topics = c(5, 15, 18), model = blog.topics,
    method = "difference", cov.value1 =
    "Liberal", cov.value2 = "Conservative",
     xlab = "More␣Conservative␣vs.␣More␣Liberal",
     xlim = c(-.1, .1), labeltype = "custom",
     custom.labels = c("Fossil␣Fuels",
     "Iraq␣War", "Clinton␣Campaign"))
```

Differences in Topic Prevalence

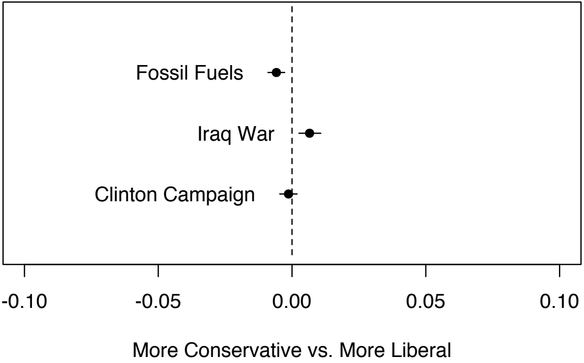


Figure 10: Ideological Differences in Topic Prevalence

What about differences over time?

```
plot(blog.effects, covariate = "day", topics = c(5, 15, 18),
     model = blog.topics, method = "continuous",
     xaxt = "n", xlab = "Year:␣2008", printlegend = F,
     font.lab = 2)
monthseq <- seq(from = as.Date("2008-01-01"),
                to = as.Date("2008-12-01"), by = "month")
monthnames <- months(monthseq)
axis(1, at = as.numeric(monthseq) - min(as.numeric(monthseq)),
     labels = monthnames)
legend(250, .095, c("Fossil␣Fuels", "Iraq␣War", "Clinton␣Campaign"),
       pch = 15, col = c("red", "green", "blue"))
```

Topic Prevalence Over Time

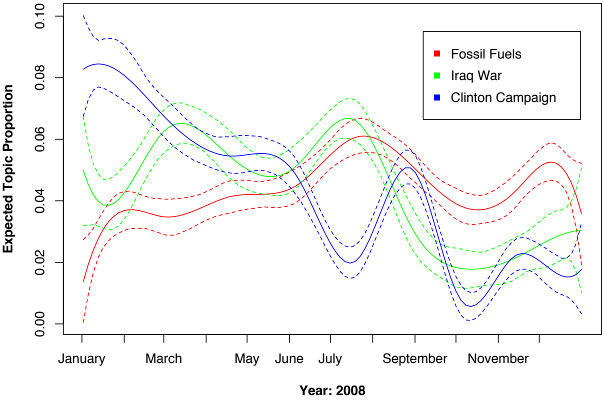


Figure 11: Differences in Topic Prevalence Over Time

Topic Prevalence Over Time

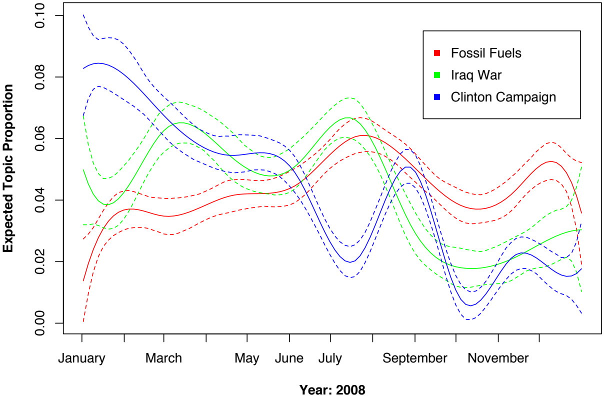


Figure 12: Differences in Topic Prevalence Over Time

Anybody want to try and interpret the plot?

Did the prevalence of the “Fossil Fuels” topic vary over time differently for liberal bloggers than it did for conservative bloggers?

```r
plot(blog.effects, covariate = "day", topics = c(5, 15, 18),
     model = blog.topics, method = "continuous",
     xaxt = "n", xlab = "Year: 2008", printlegend = F,
     font.lab = 2)
monthseq <- seq(from = as.Date("2008-01-01"),
                to = as.Date("2008-12-01"), by = "month")
monthnames <- months(monthseq)
axis(1, at = as.numeric(monthseq) - min(as.numeric(monthseq)),
     labels = monthnames)
legend(250, .095, c("Fossil Fuels", "Iraq War", "Clinton Campaign"),
       pch = 15, col = c("red", "green", "blue"))
```

Topic Prevalence Over Time, by Ideology

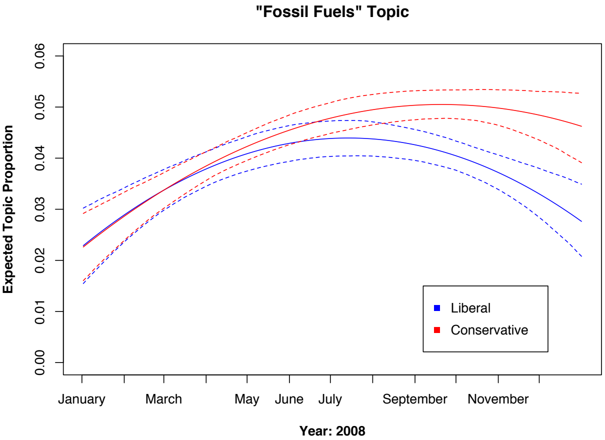


Figure 13: Differences in Topic Prevalence Over Time, by Ideology

Anybody want to try and interpret the plot?

Do liberal bloggers and conservative bloggers talk about fossil fuels differently? For these we can look at variation in *topic content*:

```
blog.topics2 <- stm(documents = docs, vocab = vocab,
    K = 20, data = meta, prevalence =˜ rating + s(day),
    content =˜ rating, seed = 123, init.type = "LDA",
    max.em.its = 75, verbose = T)
```

As before, this will take too long for class. For now, just read in this pre-estimated model:

```
blog.topics2 <- readRDS(file = "blog_topics2.rds")
```

Now we'll plot the differences:

```
#Plot content differences
plot(blog.topics2, type = "perspectives", topics = 5,
    n = 50, plabels = c("Fossil␣Fuels,\nConservative",
    "Fossil␣Fuels,\nLiberal"))
```

Talk of Fossil Fuels

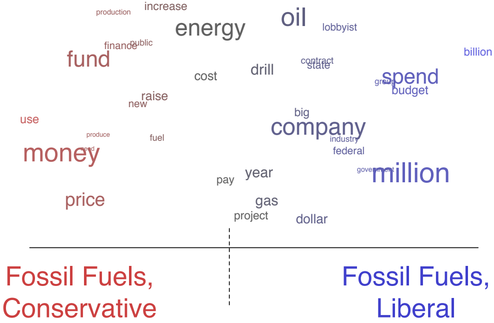


Figure 14: Differences in Talk of Fossil Fuels

Larger the word, the more frequent it is to that ideology on that topic. Farther to the edges, the more distinct the word is to that ideology when communicating that topic.

Consider another:

```
plot(blog.topics2, type = "perspectives", topics = 15,
    n = 50, plabels = c("Iraq␣War,\nConservative",
    "Iraq␣War,\nLiberal"))
```

Talk on the Iraq War

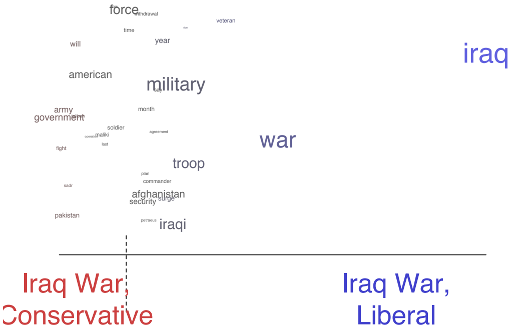


Figure 15: Differences in How the Iraq War Was Talked About

There's one more handy feature with the `stm` package: **topic correlations**.

What topics tend to co-occur together? That is, if a document engages in topic X more, what other topics is that document *also likely* to feature in a larger quantity?

```
topic.corrs <- topicCorr(blog.topics)
plot(topic.corrs)
```

Topic Correlations

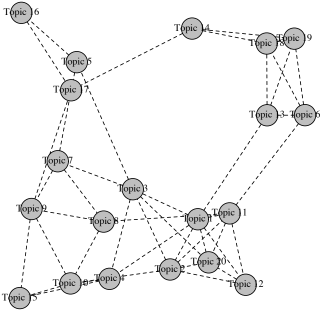
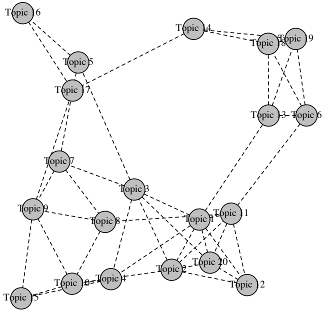


Figure 16: Topic Correlations

Topic Correlations



The “Taxes and Economy” topic (#16) tends to be co-discussed with the “Congress” topic (#17) and the “Fossil Fuels” topic (#5).

References

Bischof, Jonathan M. and Edoardo M. Airoldi. 2012. "Summarizing Topic Content with Word Frequency and Exclusivity." Pp. XX-XX in *Proceedings of the 29th International Conference on Machine Learning*. Einburgh, Scotland: Association for Computing Machinery.

Eisenstein, Jacobe and Eric Xing. 2010. *The CMU 2008 Political Blog Corpus*. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Machine Learning Department.

Farrell, Justin. 2016. "Corporate Funding and Ideological Polarization about Climate Change." *Proceedings of the National Academy of Sciences* 113(1):92-97.

Fligtein, Neil, Jonah Stuart Brundage, and Michael Schultz. 2017. "Seeing Like the Fed: Culture, Cognition, and Framing in the Failure to Anticipate the Financial Crisis of 2008." *American Sociological Review* 82(5):879-909.

Freeman, Michael K., Jason Chuang, Margaret E. Roberts, Brandon M. Stewart, and Dustin Tingley. 2015. "`stmBrowser`: Structural Topic Model Browser." R package version 1.0. Vienna, Austria: R Foundation for Statistical Computing.

References (continued)

Kuhn, Kenneth D. 2018. "Using Structural Topic Modeling to Identify Latent Topics and Trends in Aviation Incident Reports." *Transportation Research Part C: Emerging Technologies* 87:105-122.

Minmo, David, Hannah M. Wallach, Edmund Talley, Miriam Leenders, and Andrew MacCallum. 2011. "Optimizing Semantic Coherence in Topic Models." Pp. 262-272 in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland: Association for Computational Linguistics.

Roberts, Margaret E., Brandon M. Stewart, and Dustin Tingley. Forthcoming. "`stm`: R Package for Structural Topic Models." *Journal of Statistical Software*.

Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, and Edoardo M. Airoldi. 2013. "The Structural Topic Model and Applied Social Science." Presented at the NIPS 2013 Workshop on Topic Models: Computation, Application, and Evaluation, December 10, Lake Tahoe, Nevada.

Roberts, Margaret, Brandon Stewart, and Dustin Tingley. 2015. "`stm`: R Package for Structural Topic Models." R package version 1.3.3. Vienna, Austria: R Foundation for Statistical Computing.

Schmiedel, Theresa, Oliver Müller, and Jan vom Brocke. 2018. "Topic Modeling as a Strategy of Inquiry in Organizational Research: A Tutorial with an Application Example on Organizational Culture." OnlineFirst at *Organizational Research Methods*.

Taddy, Matthew A. 2012. "On Estimation and Selection of Topic Models." Pp. 1184-1193 in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics. La Palma, Canary Islands: Association for Computing Machinery.

Wallach, Hannah M., Iain Murray, Ruslan Salakhutdinov, and David Minmo. 2009. "Evaluation Methods for Topic Models."