A Brief Introduction to Topic Modeling

Marshall A. Taylor, Ph.D.

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

Introduction

- This is a mini-workshop on using NLP—specifically, topic modeling—to find latent themes and patterns in written language.
- I envision this as a tool educators might use to analyze student feedback.
- This feedback can be in the form of end-of-the-semester course evaluations, or through repeated open-ended surveys administered to students over the course of a semester where they are directly asked to (anonymously) articulate their concerns, struggles, etc., with learning during times of crisis.

Introduction

- Of course, we cannot use student feedback ad the illustrative example in this workshop. Instead, I will use State of the Union (SOTU) addresses as a simple proof of concept.
- However, I hope the application potential is clear.
- This tool will be particularly relevant for educators with large classes where sifting through individual comments may be too cumbersome and/or common discursive threads may be missed.
- All workshop materials (and more) are available on my GitHub, linked here.
- This workshop relies heavily on R and RStudio. The GitHub repository linked above includes a slide deck and R script to get you familiar with the basics of this statistical programming language.

Introduction to Topic Modeling

Topic modeling comprises a suite of tools for uncovering the **latent thematic structure** of a corpus.

Documents are presented as probabilistic mixtures of topics, so a document can display varying levels of engagement **across topics**.

We'll focus on one of the earliest and simplest forms of topic modeling: **latent Dirichlet allocation (LDA)**.

Note that to understand the mechanics behind LDA, you need to be comfortable with Bayesian statistical modeling. So we won't talk much about the math (but see Blei, Ng, and Jordan 2003 and Blei 2012 if interested).

What is LDA?

LDA is a tool for reverse engineering the latent themes—or topics—that we assume document producers tapped into when generating their texts (Blei 2012:79).

The goal is to take a corpus and decompose it into two matrices: (1) the matrix that tells us the probability that a document is in some way about a particular topic, and (2) the matrix that tells us the probability that a particular word in the corpus will appear in a particular topic.

We use the topic-word probability distribution to determine what each topic is about, and we use the document-topic probability distribution to determine which topics (or combinations of topics) are more prevalent in a document.

LDA is an unsupervised algorithm because we only provide the DTM. It iterates through the texts to estimate the probability matrices.

Conceptual Overview of LDA

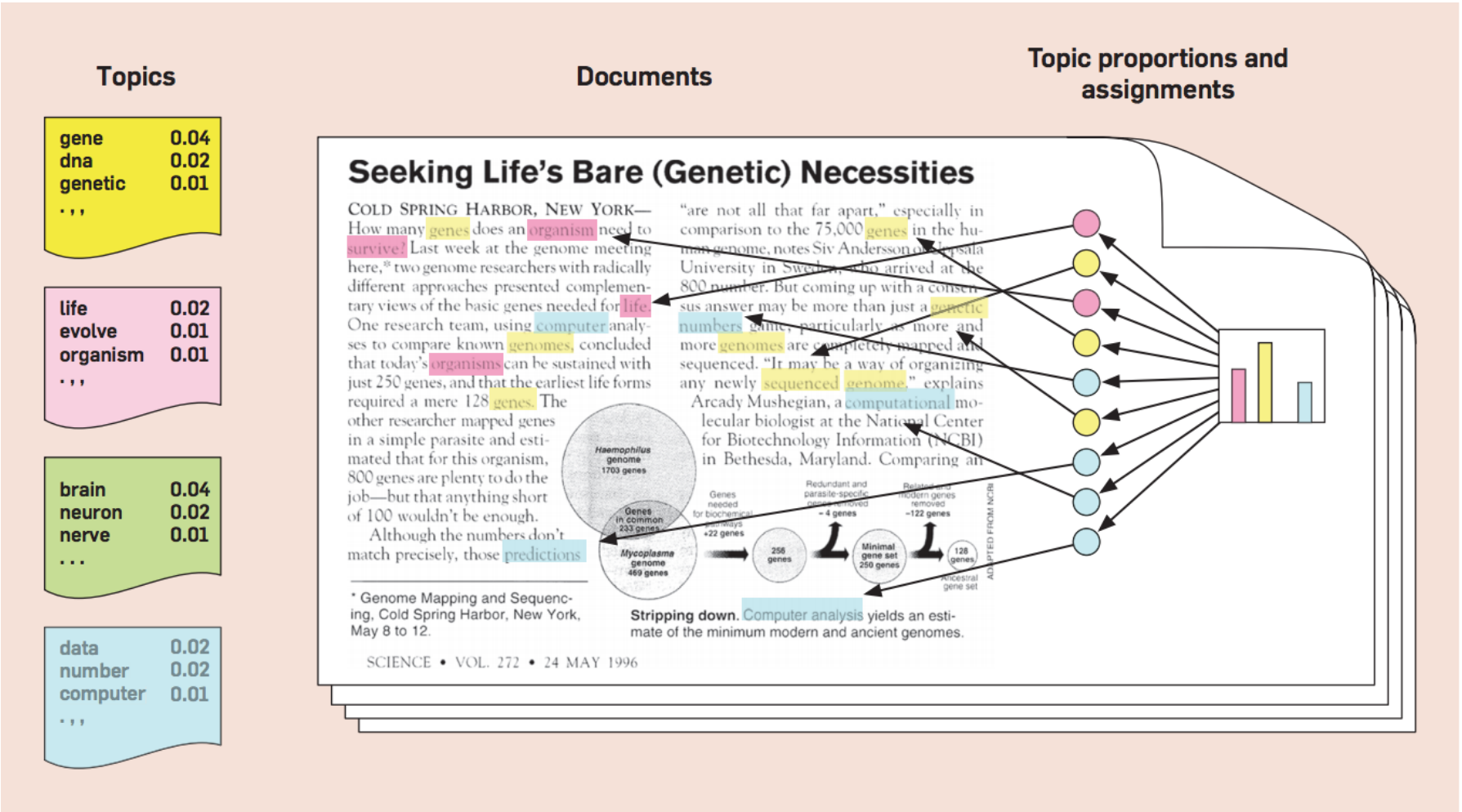

Figure 1: How LDA Works

Note: Figure from Blei (2012:78).

Conceptual Overview of LDA

Following Fligstein et al. (2017:888), imagine that you have taken a document-term matrix and converted it to a probability matrix, $P(w_i|D = d)$, where w_i is the i^{th} word in document d in the set of documents (corpus) D.

With LDA, we decompose this *observed* matrix into two *unobserved* matrices: the word-topic and document-topic probability distributions. Essentially, we estimate the set of topics k that allows us multiply these two probabilistic quantities together and get as close as possible to the observed $w_i|D = d$ probability:

$$P(w_i|D = d) = \sum_{k=1}^{K} P(w_i|z_i = k) \times P(z_i = k|D = d) \tag{1}$$

Where z_i is the topic assignment of the i^{th} word w in document d in corpus D.

LDA Interpretation

The first step in interpreting LDA output is usually to take a look at the words that "load" the highest on each topic: that is, the words with the highest probability of being associated with each topic.

Consider this example from an LDA analysis of Associated Press articles:

LDA Interpretation

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Figure 2: How LDA Works

Note: Figure from Blei et al. (2003:1009).

LDA Interpretation

Importantly, LDA models are "mixture models" (Blei et al. 2003). This means that a word is not assigned to one and only one topic, but instead a vector of probabilities across all topics. The same applies to the document-topic probabilities: a document is assigned a probability for each topic.

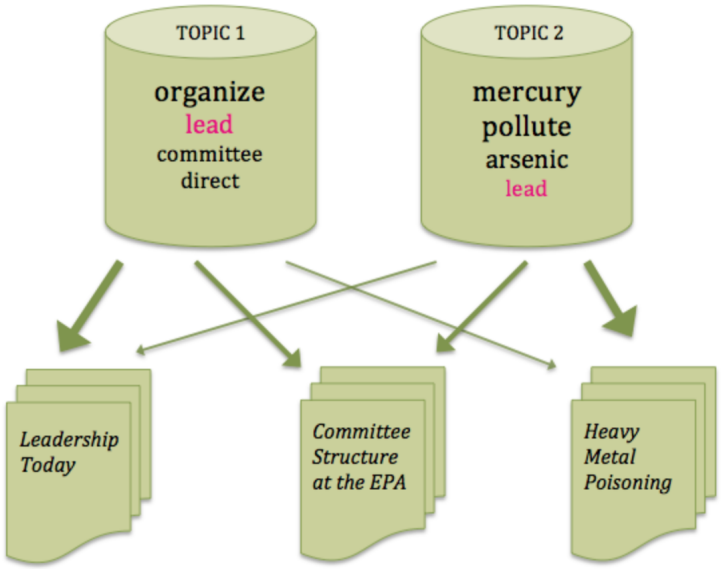

Figure 3: Words aren't Exclusive to Topics

Note: Figure from Underwood (2012).

Let's work through an example with the SOTUs from George H. W. Bush in 1989 to Donald Trump in 2019 (Benoit and Watanabe 2019).

```
install.packages(c("tm", "topicmodels", "ggplot2",
                   "quanteda.corpora", "reshape2",
                   "tidytext", "dplyr", "textstem",
                   "textreg", "LDAvis", "servr"))
library(tm)
library(topicmodels)
library(ggplot2)
library(quanteda.corpora)
library(reshape2)
library(tidytext)
library(dplyr)
library(textstem)
library(textreg)
library(LDAvis)
library(servr)

load("data_corpus_sotu.rda")
sotus <- data_corpus_sotu$documents
sotus <- sotus[which(sotus$Date>="1989-02-09"),]
```

Let's remove punctuation, any non-ASCII characters, capitalization, numbers, English stop words, and excess white space. Then we'll lemmatize the corpus and remove some sparse terms to get a DTM with 31 speeches, 2,402 unique words, and 84,594 total words.

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

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

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

sotus.tm <- VCorpus(VectorSource(sotus.tm))

sotus.dtm <- DocumentTermMatrix(sotus.tm)
sotus.dtm <- removeSparseTerms(sotus.dtm, .9)
```

So what are the latent themes in this SOTU corpus?

We'll estimate the topics using the `topicmodels` package (Grün and Hornik 2011).

Similar to hierarchical clustering algorithms, we have to supply the number of topics, k, that we believe are in the corpus. I have no reason to believe there is a specific number of topics here, so let's start arbitrarily with 10 and see what the topics look like:

```
#Estimate some topics
sotus.topics.10 <- LDA(sotus.dtm, k = 10,
    control = list(seed = 567))
```

What are the top 10 terms associated with each topic?

```
terms(sotus.topics.10, 10)
```

For example, if you were to pluck a word at random from topic #9, the word with the highest probability of being chosen is “people.”

Let's visualize each topic's top terms and their probabilities (code adapted from Bail [2018] and Jackson [2016]):

```
top.terms <- sotus.topics.10@beta
  colnames(top.terms) <- sotus.topics.10@terms
  top.terms <- exp(top.terms)
  top.terms <- t(top.terms)
  colnames(top.terms) <- paste("topic_", 1:10, sep = "")

  top.terms <- melt(top.terms)
  top.terms <- top.terms %>%
    group_by(Var2) %>%
    top_n(10, value) %>%
    ungroup() %>%
    arrange(Var2, -value) %>%
    mutate(order = row_number())

  ggplot(top.terms, aes(x = rev(order), y = value, fill = Var2)) +
    geom_bar(stat = "identity", color = "black") +
    ylim(0, .03) +
    xlab("") + ylab("") +
    facet_wrap(~Var2, scales = "free") +
    coord_flip() +
    guides(fill = F) +
    theme_bw() +
    scale_x_continuous(breaks = rev(top.terms$order),
                       labels = top.terms$Var1,
                       expand = c(0,0))
```

The Plot

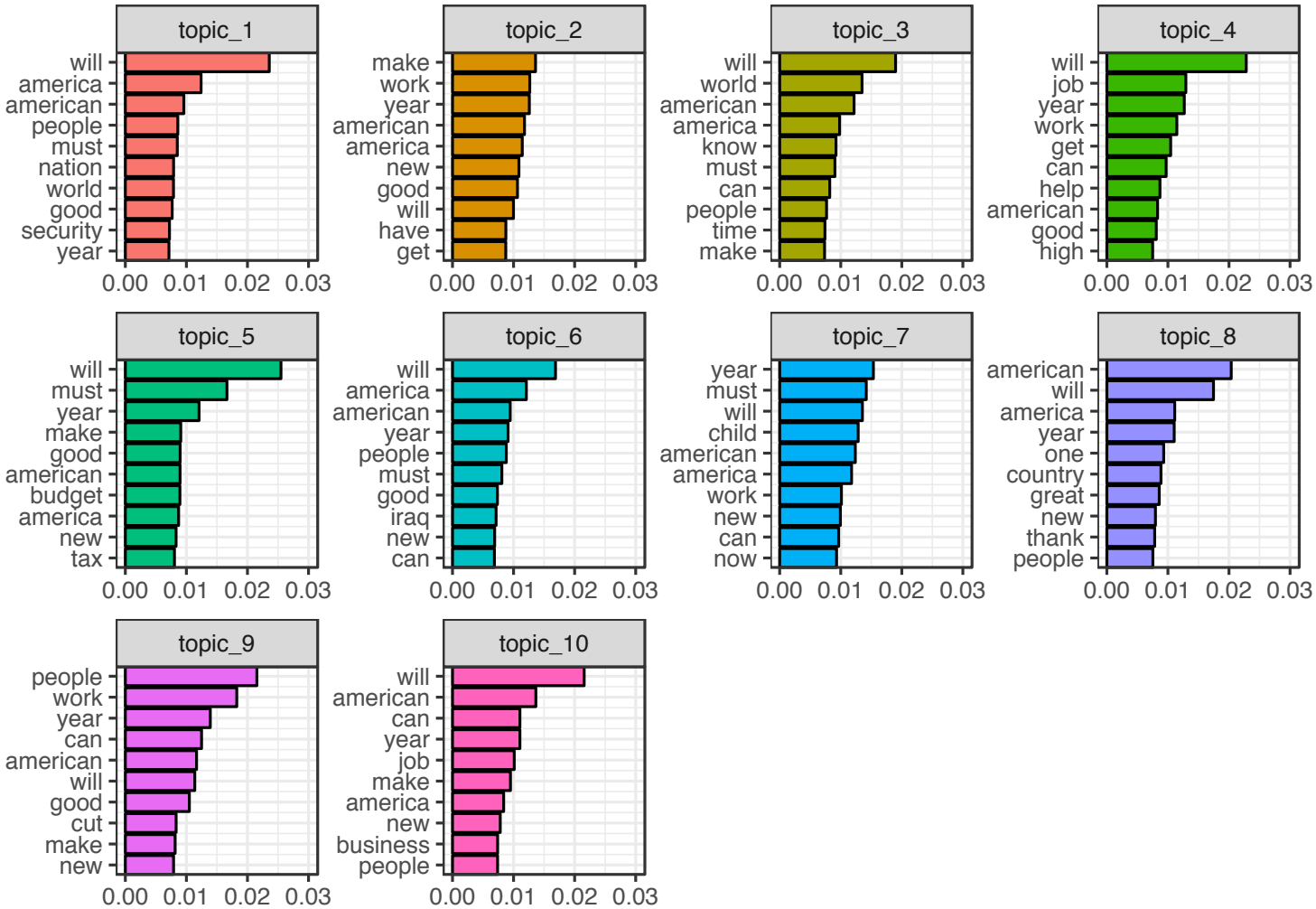

Figure 4: Top Words per 10 SOTU Topics

Interpretation

The next step is to assess the face validity of the topics.

Do they make sense? Do some seem too specific? Too general? To "messy"?

If they tend to be too general, then you may want to increase the number of topics. If they tend to be too specific, then maybe you want to decrease the number of topics.

Next week we'll talk about some statistical techniques for helping to determine the number of topics. **However**, nothing supplants the importance of having topics that make sense to the human eye!

Let's try a five-topic solution:

```
#Estimate a 5-topic solution
sotus.topics.5 <- LDA(sotus.dtm, k = 5, control = list(seed = 567))

  #Visualize the word distributions for top words per topic
  top.terms <- sotus.topics.5@beta
  colnames(top.terms) <- sotus.topics.5@terms
  top.terms <- exp(top.terms)
  top.terms <- t(top.terms)
  colnames(top.terms) <- paste("topic_", 1:5, sep = "")

  top.terms <- melt(top.terms)
  top.terms <- top.terms %>%
    group_by(Var2) %>%
    top_n(10, value) %>%
    ungroup() %>%
    arrange(Var2, -value) %>%
    mutate(order = row_number())

  ggplot(top.terms, aes(x = rev(order), y = value, fill = Var2)) +
    geom_bar(stat = "identity", color = "black") +
    ylim(0, .03) +
    xlab("") + ylab("") +
    facet_wrap(~Var2, scales = "free") +
    coord_flip() +
    guides(fill = F) +
    theme_bw() +
    scale_x_continuous(breaks = rev(top.terms$order),
                       labels = top.terms$Var1,
                       expand = c(0,0))
```

The Plot

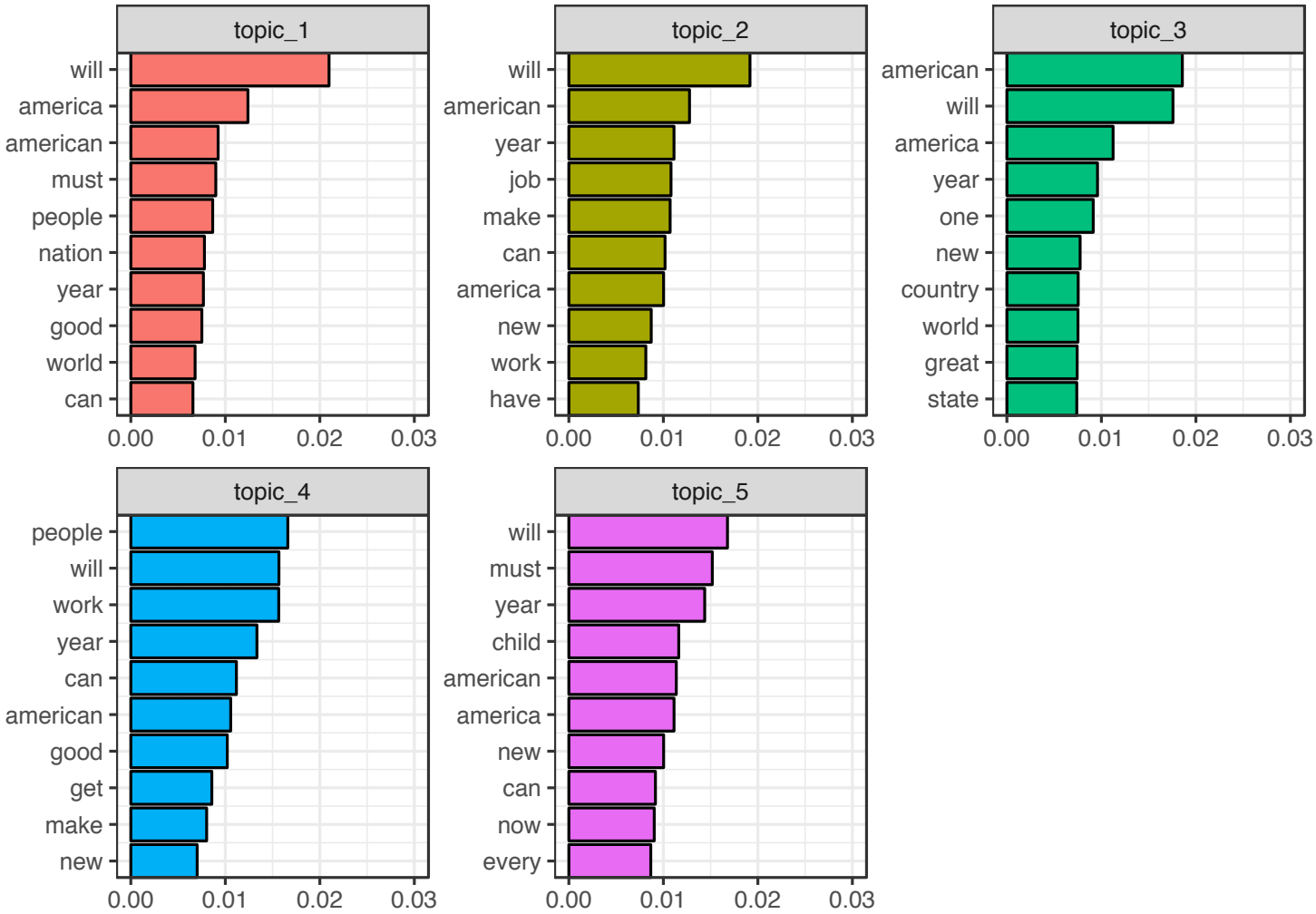

Figure 5: Top Words per 5 SOTU Topics

Let's go in the opposite direction and try 20:

```
#Estimate a 20-topic solution
sotus.topics.5 <- LDA(sotus.dtm, k = 20, control = list(seed = 567))

  #Visualize the word distributions for top words per topic
  top.terms <- sotus.topics.20@beta
  colnames(top.terms) <- sotus.topics.20@terms
  top.terms <- exp(top.terms)
  top.terms <- t(top.terms)
  colnames(top.terms) <- paste("topic_", 1:20, sep = "")

  top.terms <- melt(top.terms)
  top.terms <- top.terms %>%
    group_by(Var2) %>%
    top_n(10, value) %>%
    ungroup() %>%
    arrange(Var2, -value) %>%
    mutate(order = row_number())

  ggplot(top.terms, aes(x = rev(order), y = value, fill = Var2)) +
    geom_bar(stat = "identity", color = "black") +
    ylim(0, .04) +
    xlab("") + ylab("") +
    facet_wrap(~Var2, scales = "free") +
    coord_flip() +
    guides(fill = F) +
    theme_bw() +
    scale_x_continuous(breaks = rev(top.terms$order),
                       labels = top.terms$Var1,
                       expand = c(0,0))
```

The Plot

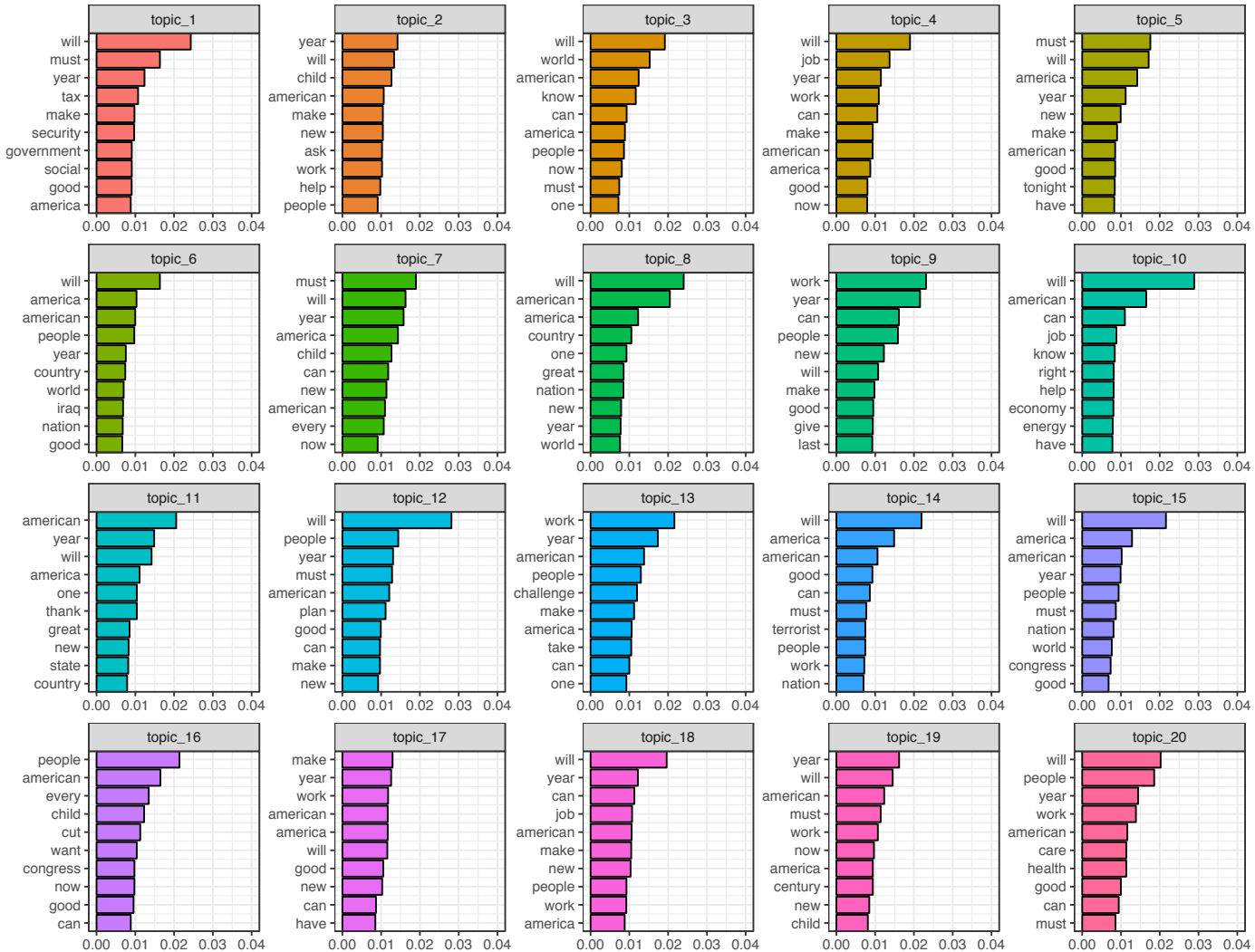

Figure 6: Top Words per 20 SOTU Topics

Keep Iterating!

So 20 topics seem too fine-grained, but 5 topics seem too general.

Let's go back to 10 topics. But "america," "american," and some of these modal verbs occur so often and across topics to not be informative. Let's remove some of them by re-generating the DTM, but this time adding `c(stopwords("english"), "will", "can", "must", "america", "american", "year", "people")` to the `removeWords` preprocessor.

The Plot

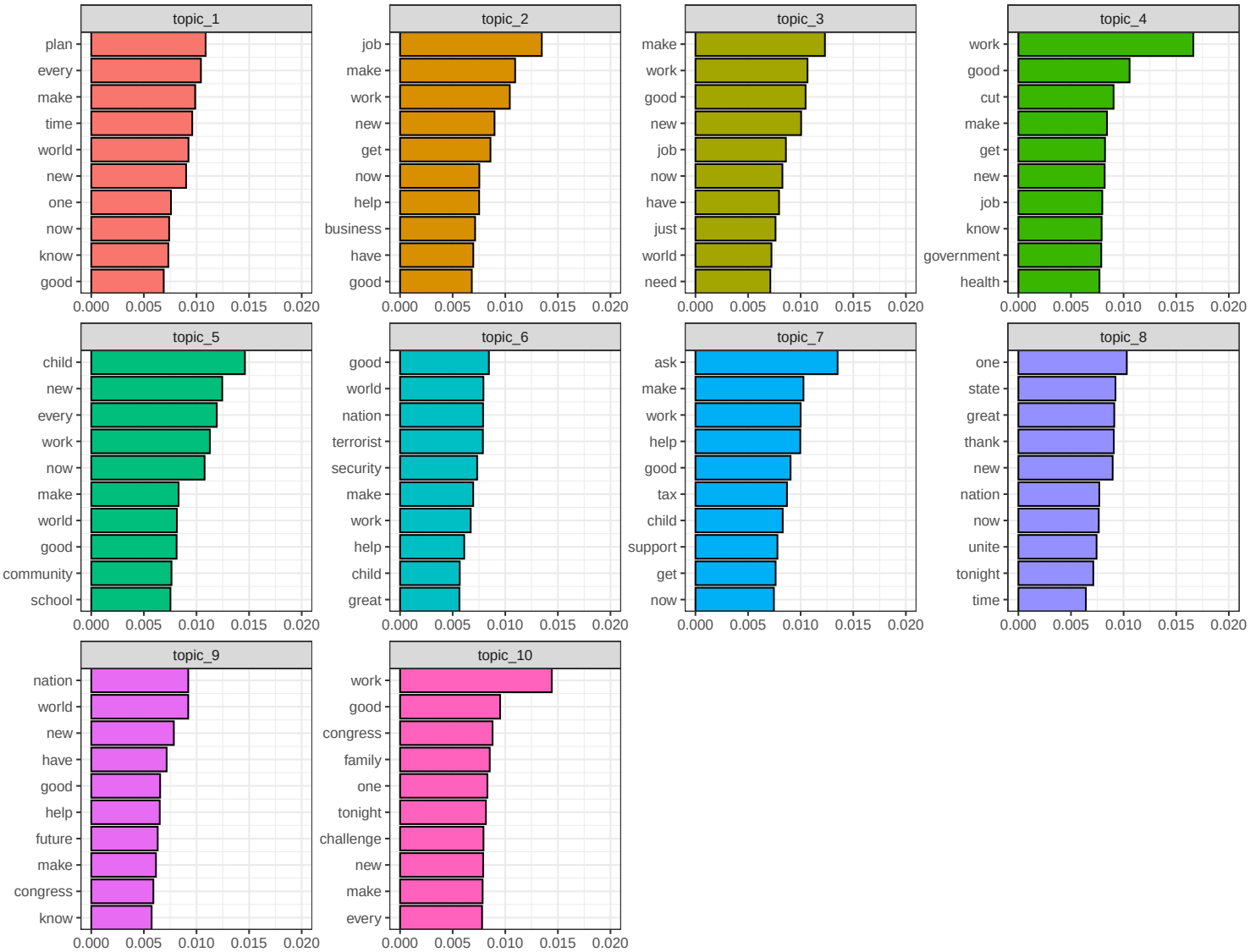

Figure 7: Top Words per 10 SOTU Topics with more Preprocessing

Interpretation, Again

It seems to me that a 10-topic solution still gives us some amorphous topics.

After playing around with different k, I think a 6-topic solution with those super frequent terms removed looks ok:

The Plot

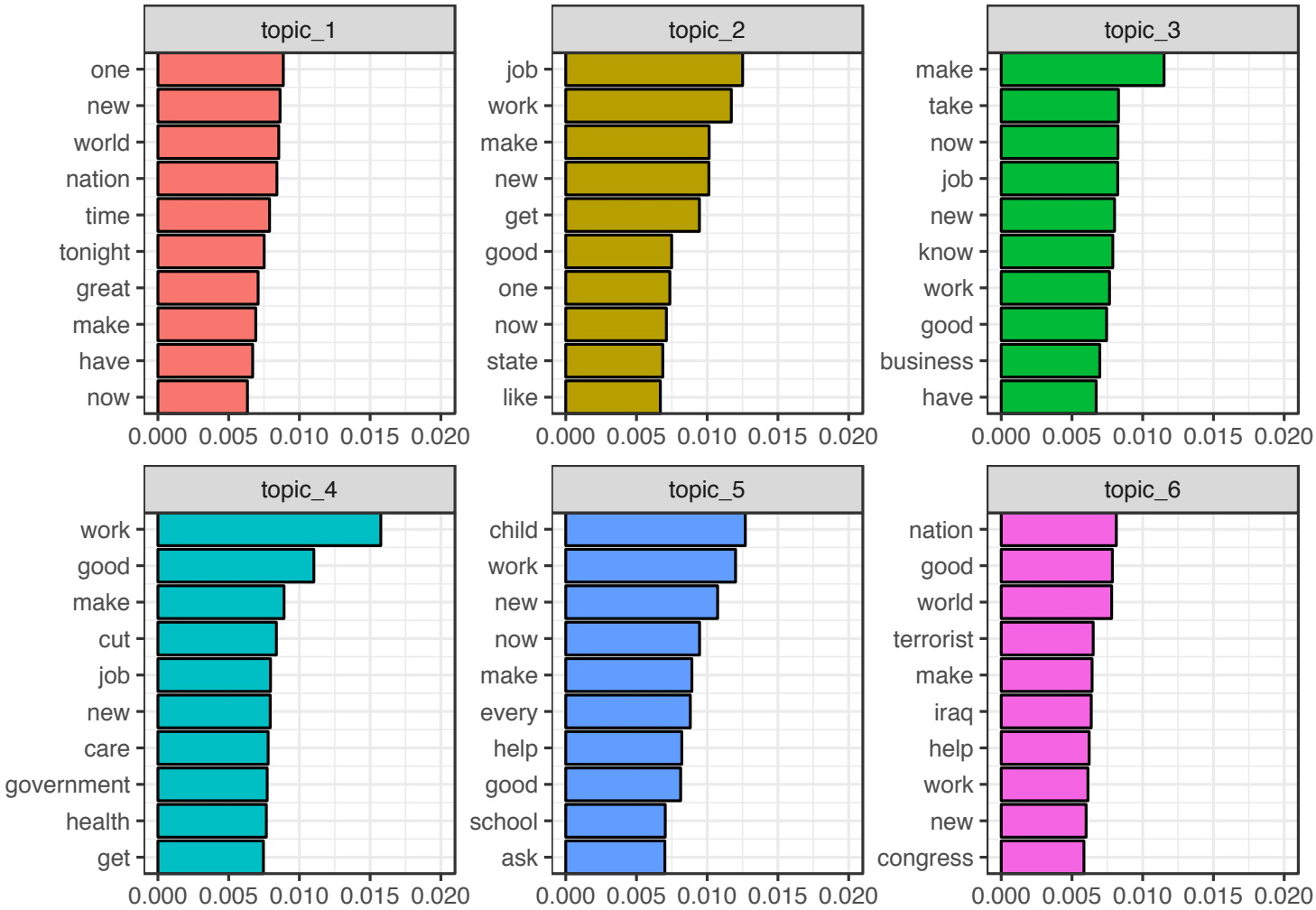

Figure 8: Top Words per 6 SOTU Topics

Specific Topics

For instance, topic #6 seems to be a "War on Terror" topic.

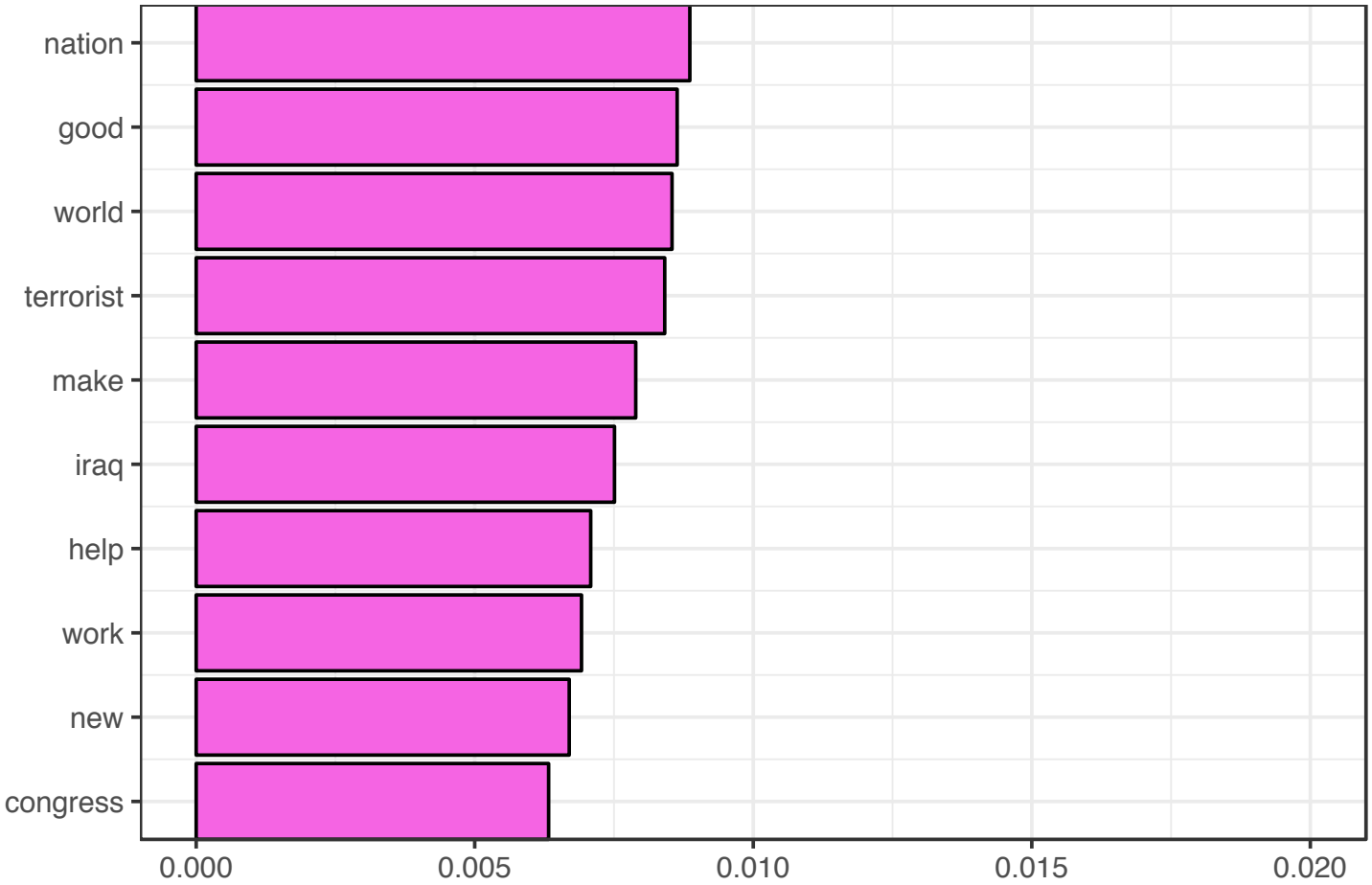

Figure 9: A "War on Terror" Topic

Specific Topics

And maybe topic #4 is a "Government and Employment Healthcare" topic.

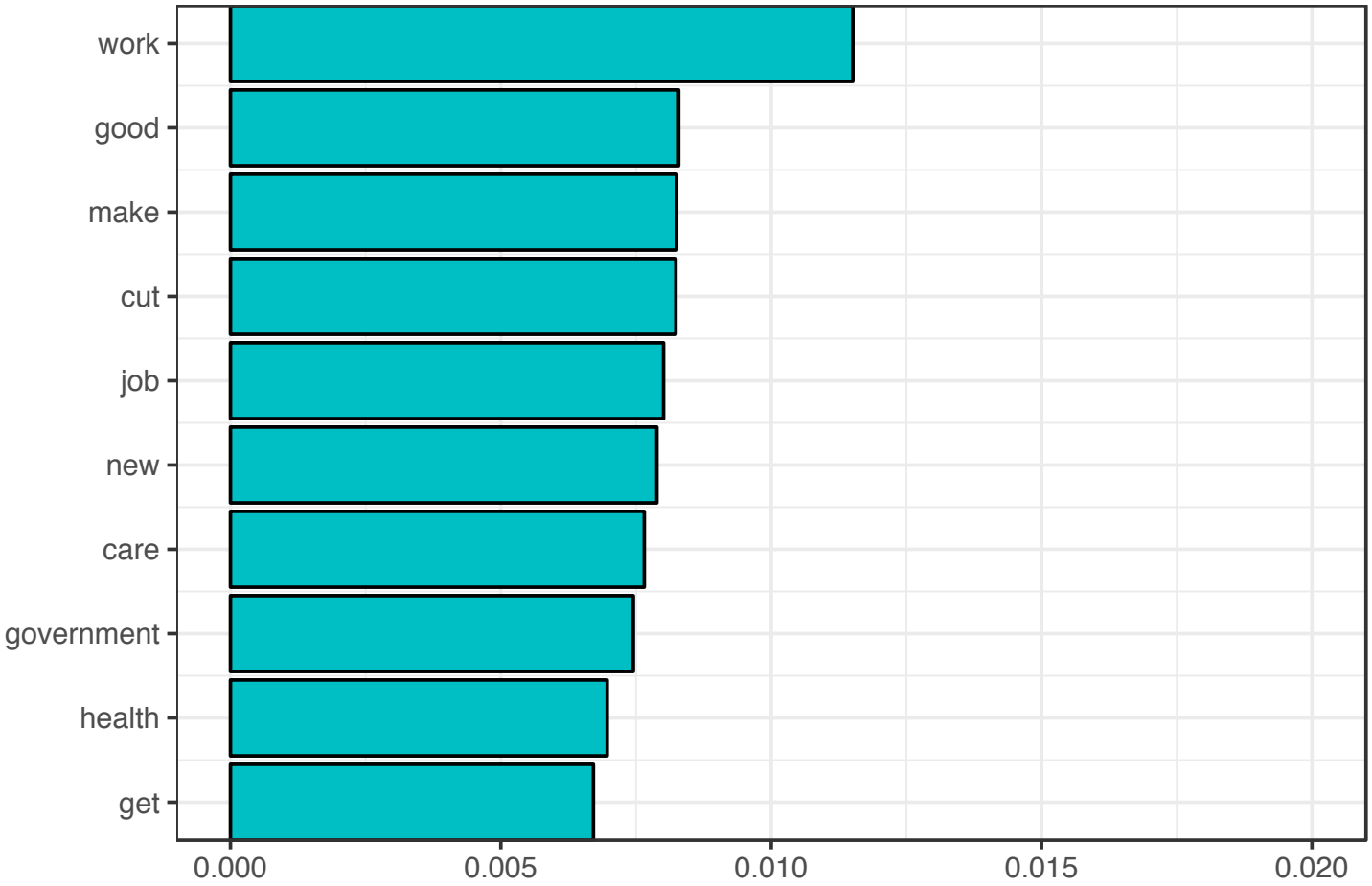

Figure 10: A "War on Terror" Topic

Specific Topics

Topic #5 seems to clearly be a "Children and Schooling" topic.

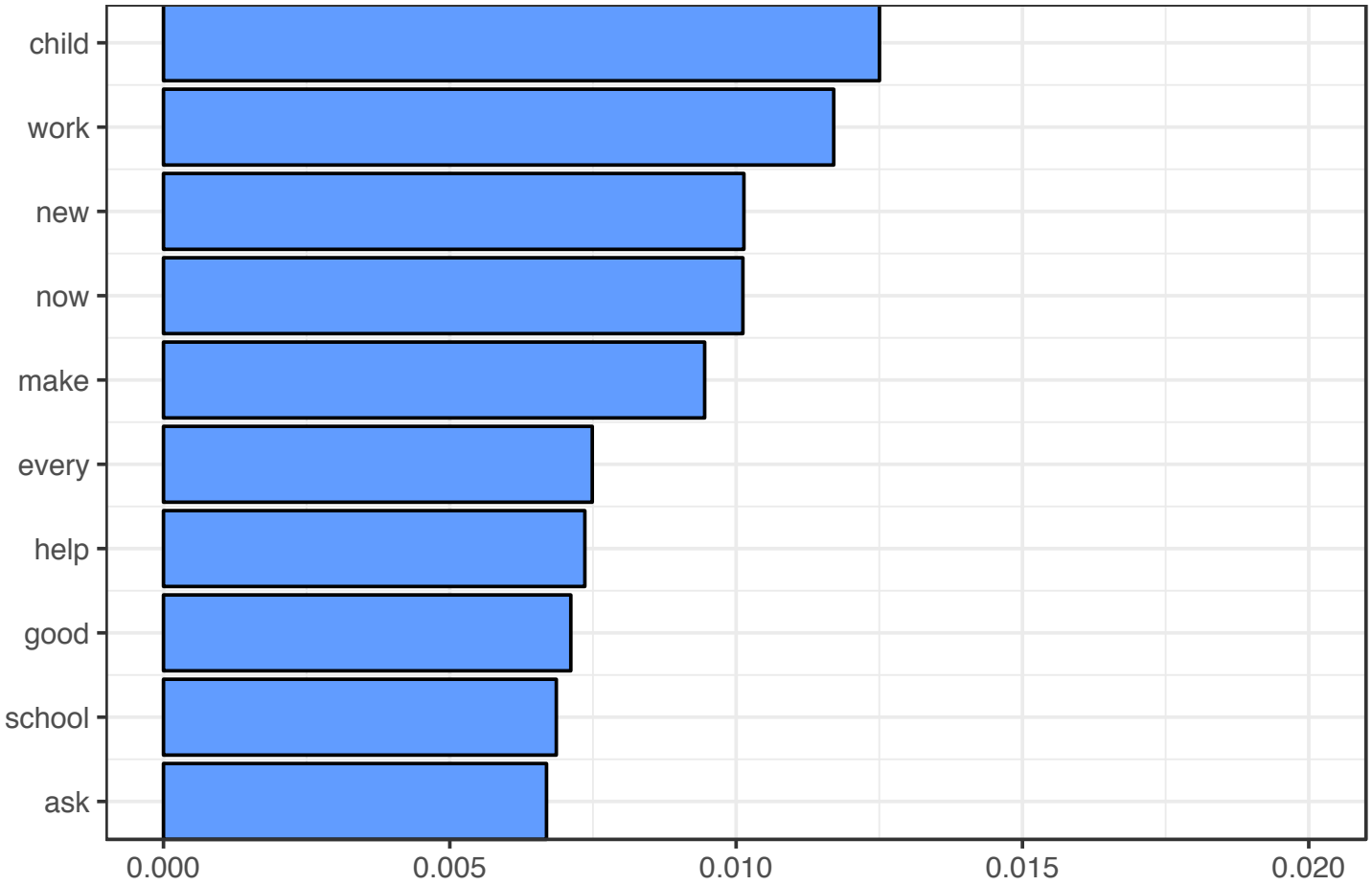

Figure 11: A "Children and Schooling" Topic

Distinctive Words

We can also look at the **most distinctive** terms associated with each topic to help interpretation.

A useful metric for doing this is called **lift** (Taddy 2012). The terms we have been looking at are the terms with the **highest probability** of being associated with the topic. When we look at terms' lift scores, we are looking at the terms **most distinctive** to the topic.

Word w_i's lift in topic k, l_{ik}, is defined as (Sievert and Shirley 2014:65):

$$l_{ik} = \frac{P(w_i|z_i = k)}{P(w_i)} \tag{2}$$

Lift is the ratio of term w_i's probability in topic k to term i's marginal probability across all documents (Sievert and Shirley 2014:65). Larger values indicate that term w_i is mostly accounted for by that topic—it is a word distinct to that topic.

Term Relevance

Luckily, we don't have to have to think about a term's lift in a topic separately from its raw topic-specific probability.

Sievert and Shirley (2014:66) blend these two metrics together with what they call **term relevance**. This is the weighted product of the term's probability and it's lift:

$$r_{ik} = \lambda P(w_i|z_i = k) + \left((1-\lambda)\frac{P(w_i|z_i = k)}{P(w_i)}\right) \tag{3}$$

The weighting factor, λ, adjusts how much weight we want to place on probability over lift. Higher the λ, the more we define term relevance in terms of high probability rather than distinctiveness.

Sievert and Shirley (2015) have a great package for visualizing term relevance and interpreting the topics: `LDAvis`.

Let's give it a shot:

```
#Interactive visualization with high prob and lift
json <- createJSON(phi = exp(sotus.topics.6@beta),
                   theta = sotus.topics.6@gamma,
                   doc.length = rowSums(
                    as.matrix(sotus.dtm)),
                   term.frequency = colSums(
                    as.matrix(sotus.dtm)),
                   vocab = colnames(
                    as.matrix(sotus.dtm)))
serVis(json, out.dir = "vis",
    open.browser = interactive())
```

Click the little "open in new window" button in the Viewer pane to open the visualization in your web browser.

LDAvis Output

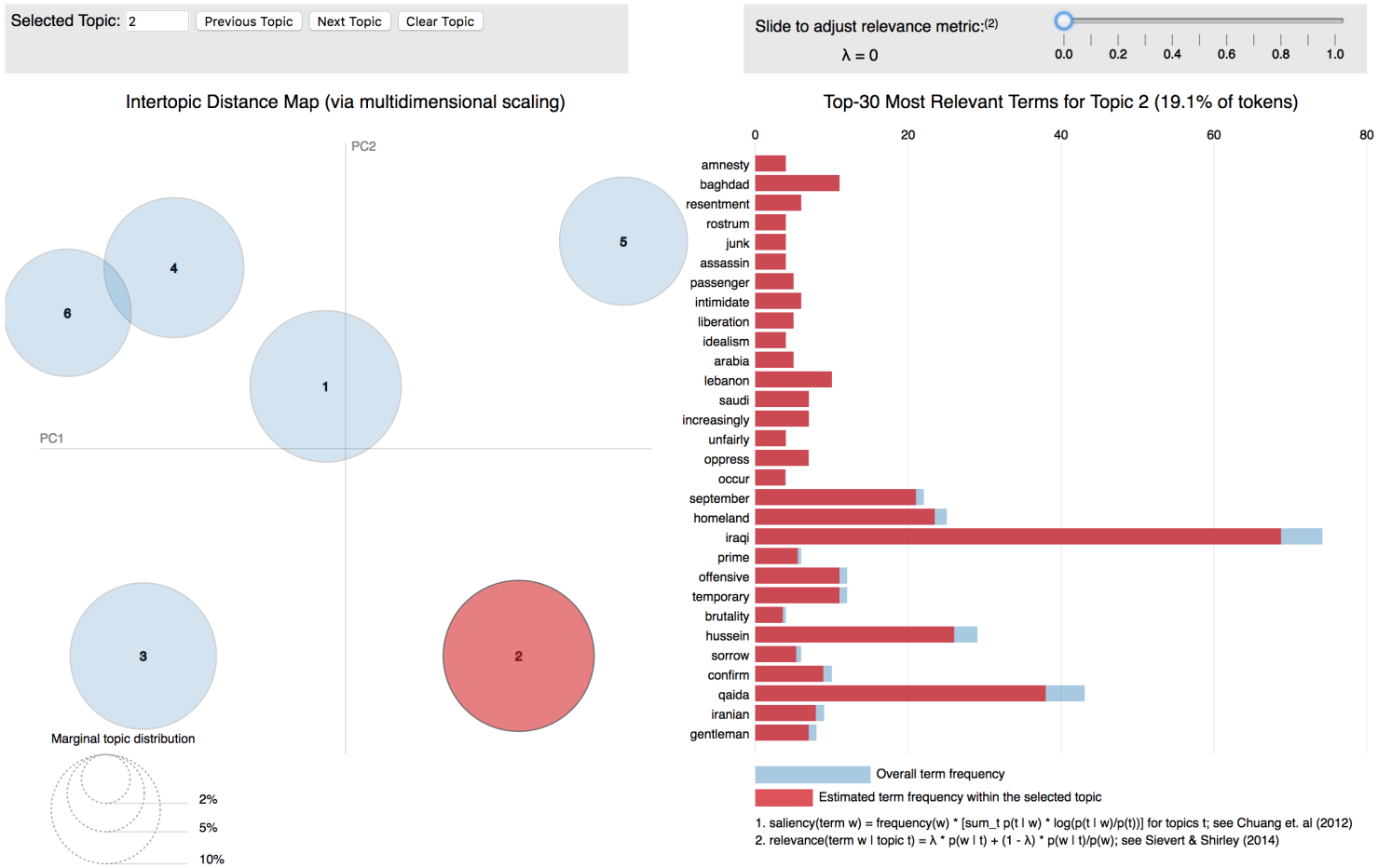

Figure 12: Screenshot from `LDAvis` on SOTU Speeches

LDAvis Output

The left-hand side shows the first two dimensions of what's called a *principal components analysis* on the document-topic probability distribution.

Basically, the closer together two topics are, the more thematically similar they are.

"Final" Interpretation

After playing around with the visualization, it seems to me that, going by the order of the topics in the bar charts (rather than the order in the visualization), that:

- **Topic #1 ≈ "One Great Nation"**
- **Topic #2 ≈ "New Jobs"**
- **Topic #3 ≈ "State of the Economy"**
- **Topic #4 ≈ "Welfare State"**
- **Topic #5 ≈ "Children and Schooling"**
- **Topic #6 ≈ "War on Terror"**

Notice that the "Children and Schooling," "Welfare State," and "State of the Economy" topics are close together in the principal components space. Makes sense, right? They are all similar in the sense that they deal with issues of the state.

Now that we have the topics more or less interpreted, we can look to see how engagement with these topics varies across SOTUs.

```
#Top topics per SOTU
topics(sotus.topics.6, 1)
```

Now run the heat map code underneath it to get...

The Plot

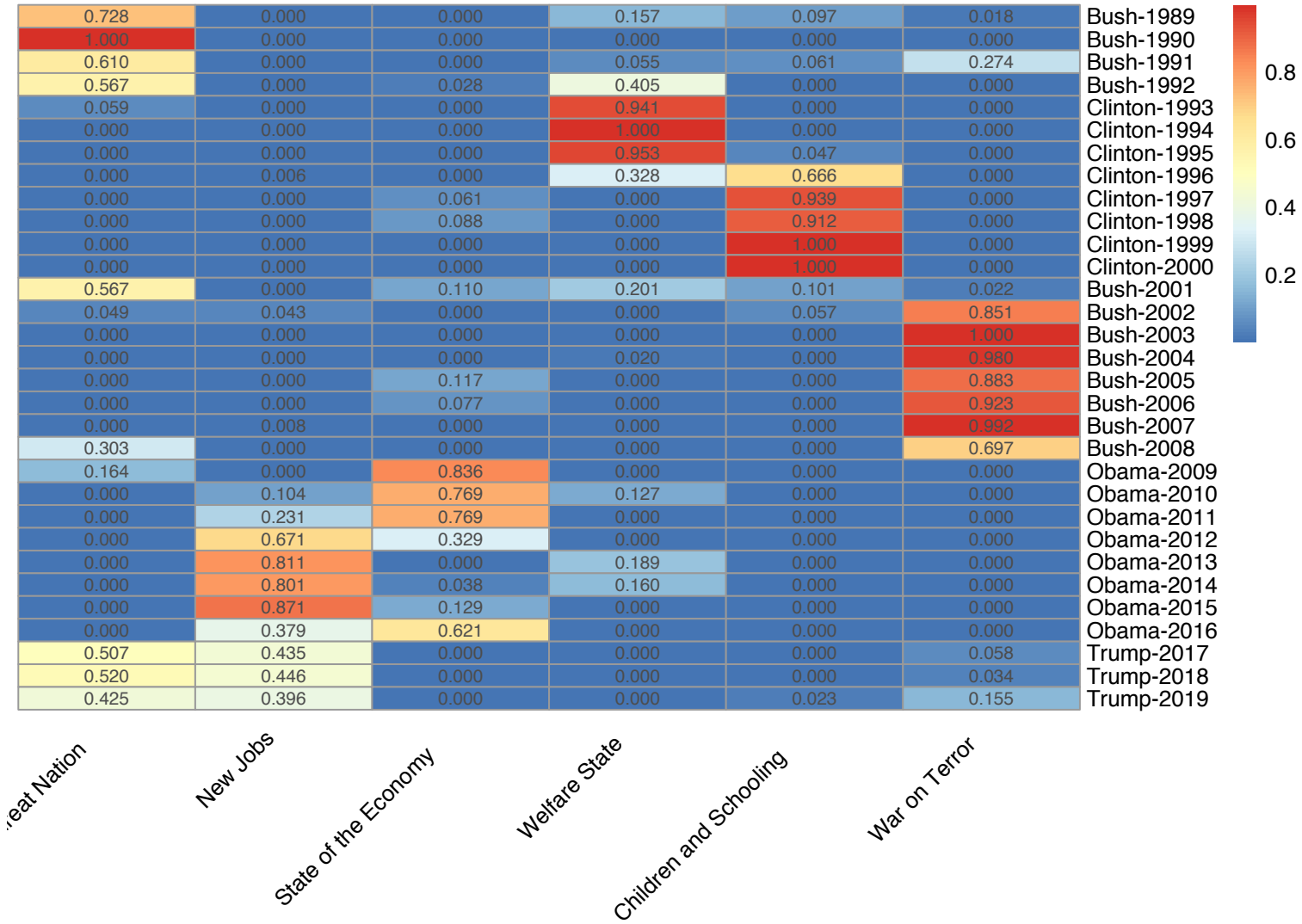

Figure 13: Document-Topic Heatmap

What did topic engagement look like over time?

```
sotus <- cbind(sotus, doc.topic)

  sotus.time <- melt(sotus, measure.vars = c("One Great Nation", "New Jobs",
                                             "State of the Economy",
                                             "Welfare State", "Children and Schooling",
                                             "War on Terror"),
                     id.vars = "Date")
  sotus.time$year <- substring(sotus.time$Date, 1, 4)

  ggplot(sotus.time, aes(x = year, y = value, fill = variable)) +
    geom_bar(stat = "identity", color = "black") +
    ylab("Proportion of Speech Engaged with the Topic") +
    xlab("Date") +
    scale_x_discrete(breaks = seq(1989, 2019, 4)) +
    theme_bw() +
    theme(legend.position = "bottom",
          axis.title = element_text(face = "bold"),
          legend.title = element_text(face = "bold")) +
    guides(fill = guide_legend(title = "Topic"))
```

The Plot

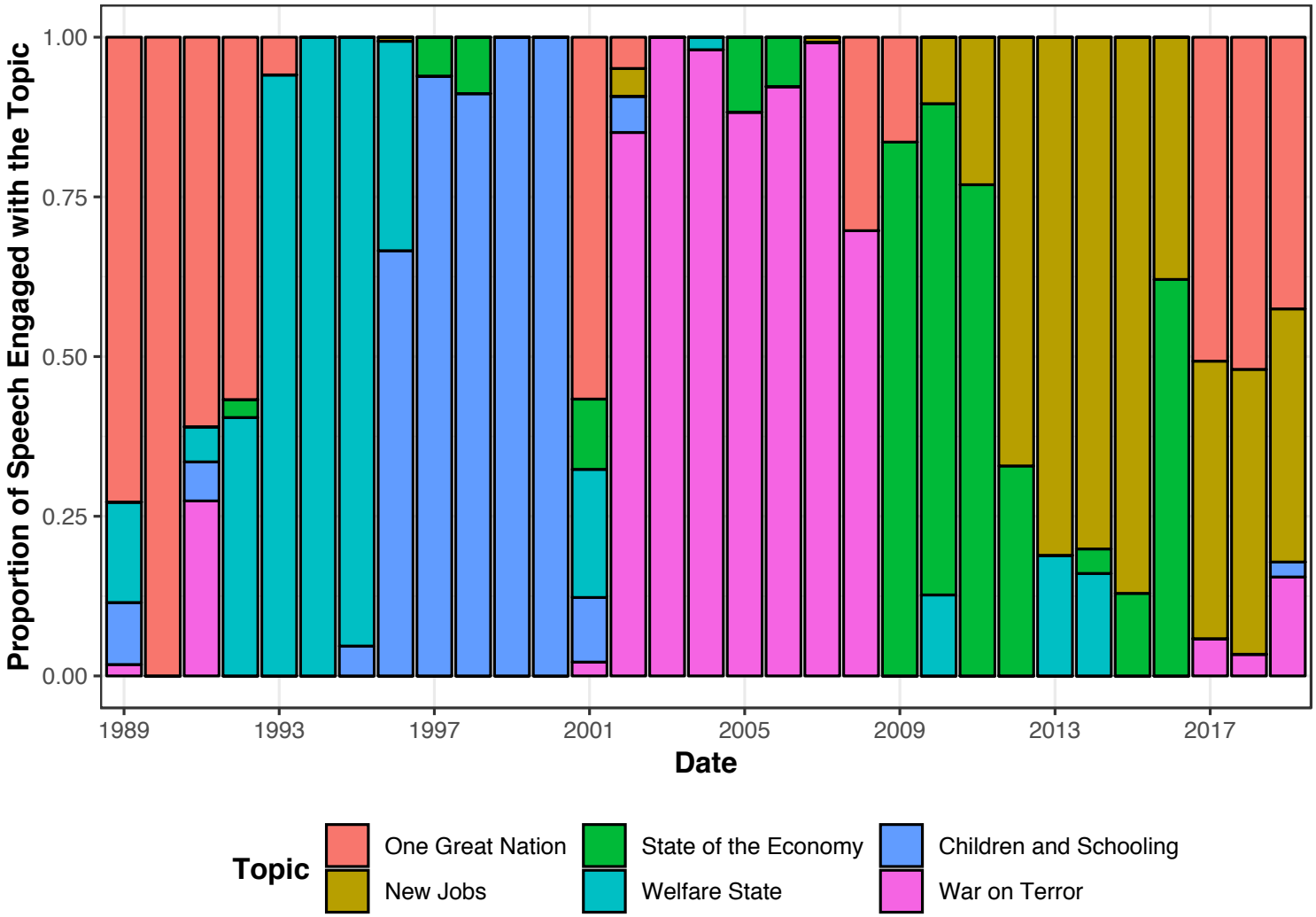

Figure 14: Topic Engagement Over Time

Conclusion

While I wish I could have illustrated topic modeling using student feedback data, I hope this this brief SOTU example has drawn your attention to how topic modeling might be useful in summarizing student feedback.

I have used topic modeling in this way, where I used it to find the underlying strengths and weaknesses that students articulated about their courses across a corpus of over 100,000 feedback documents.

Conclusion

Here's one way that I foresee instructors using this technique:

- Invite students to anonymously articulate the strengths and weaknesses of a class and their concerns about their learning experiences using open-ended questions through simple web-based survey platforms such as SurveyMonkey or Google Forms.
- The surveys could be open throughout the semester, so students can respond "in real time" and as thoughts arise.
- The instructor could then use LDA to identify the strengths, weaknesses, and concerns as expressed by students.
- The instructor could use this information to adjust their pedagogy and/or materials as the semester unfolds, share the general trends with their peers, or use findings to adjust course standards/goals in future classes.
- The "real time" nature of incoming data and the ease and speediness with which instructors can derive themes also means that instructors might be able to assess how the strengths, weaknesses, and concerns change with external exigencies—such as stay-at-home orders, internet difficulties, changes to university scheduling, etc.

Conclusion

Topic models have become very popular in social science research. They can be very useful for identifying themes, especially in large corpora. However, researchers often over-estimate what topic modeling can give them.

Topic models can be *very* sensitive to model specification. What this means is that changes to the pre-modeled data—say, removing sparse terms with a different sparsity factor, or changing the number of topics—can produce sometimes radically different topic estimates.

The workshop repository on my GitHub contains a slide deck and R script for another method called **structural topic modeling**, which you can use to help guide you search for an optimal k (as well as regress your topics on student, course, and instructor covariates if that data are relevant and available). But the truth is that there is no magic number. Topic modeling involves a lot of iteration between topic estimation and interpretation.

References

Bail, Christopher A. 2018. "Topic Modeling." Retrieved February 25, 2019 (link).

Benoit, Kenneth and Kohei Watanabe. 2019. "`quanteda.corpora`: A Collection of Corpora for Quanteda." R package version 0.86. Retrieved February 19, 2019 (link).

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3:993-1022.

Blei, David M. 2012. "Probabilistic Topic Models." *Communications of the ACM* 55(4):77-84.

Fligstein, 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.

Grün, Bettina and Kurt Hornik. 2011. "`topicmodels`: An R Package for Fitting Topic Models." *Journal of Statistical Software* 40(13):1-30.

Jackson, Simon. 2016. "Ordering Categories within `ggplot2` Facets." *Blogr*. Retrieved February 25, 2019 (link).

Sievert, Carson and Kenneth E. Shirley. 2014. "`LDAvis`: A Method for Visualizing and Interpreting Topics." Pp. 63-70 in Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. Baltimore, MD: Association for Computational Linguistics.

Sievert, Carson and Kenny Shirley. 2015. "`LDAvis`: Interactive Visualization of Topic Models." R package version 0.3.2. Retrieved February 25, 2019 (link).

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: Journal of Machine Learning Research.