Campaigns and Elementary Text Analysis

GOV 1347 Lab: Week XII

Matthew E. Dardet

Harvard University

December 4, 2024

Check-In

Any questions? Ponderings? Holiday recollections?

Agenda

- Processing Text Data
- Methods for Text Content Analysis
    - Topic Models
    - LLMs

Section 1

Processing Text Data

Quantitative Text Analysis Basics

- Using document-feature matrices to represent corpora of text as data.

Quantitative Text Analysis Basics

- Using document-feature matrices to represent corpora of text as data.
- Visualizing quantitative representations of text using word clouds, keyness plots, and feature correspondence network plots.

Quantitative Text Analysis Basics

- Using document-feature matrices to represent corpora of text as data.
- Visualizing quantitative representations of text using word clouds, keyness plots, and feature correspondence network plots.
- How to ultimately apply textual data analysis to simplify the intensive process of manual coding undertaken by Vavreck (2009) in her work on the 2008 election.

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.
- But, *quantitative* analysis is useful when:

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.
- But, *quantitative* analysis is useful when:
  - *Distant reading* (compared to *close reading*) is sufficient.

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.
- But, *quantitative* analysis is useful when:
  - *Distant reading* (compared to *close reading*) is sufficient.
  - When the size of texts/documents (corpus) to analyze is huge.

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.
- But, *quantitative* analysis is useful when:
  - *Distant reading* (compared to *close reading*) is sufficient.
  - When the size of texts/documents (corpus) to analyze is huge.
  - When we are pursuing objective and reproducible scientific results.

Why to Analyze Texts Quantitatively

- Text analysis is frequently qualitative.
- But, *quantitative* analysis is useful when:
  - *Distant reading* (compared to *close reading*) is sufficient.
  - When the size of texts/documents (corpus) to analyze is huge.
  - When we are pursuing objective and reproducible scientific results.

How to Analyze Texts Quantitatively

- *Reduce complexity:* Language is extraordinarily complex, with subtlety and nuance. We need to represent documents as straightforward mathematical objects.

How to Analyze Texts Quantitatively

- *Reduce complexity:* Language is extraordinarily complex, with subtlety and nuance. We need to represent documents as straightforward mathematical objects.
- (Traditional) *Pre-processing:* What can be simplified? Which complexity can be removed?
    - Tokenize (using whitespace)
    - Remove grammatical structure: *bag of words* assumption
    - Remove punctuation
    - Remove capitalization
    - Remove stop words (e.g., a, it, the, would . . .)
    - Stemming (e.g., radicalize, radical $\rightsquigarrow$ radic)

Using Quanteda in R

```
harris_1028_tokens_processed <- tokens(harris_1028_enos$text,
                                       remove_symbols = TRUE,
                                       remove_numbers = TRUE,
                                       remove_punct = TRUE,
                                       remove_separators = TRUE) |>
  tokens_tolower() |>
  tokens_remove(pattern = c("joe", "biden", "donald", "trump", "president",
                           "kamala", "harris")) |>
  tokens_remove(pattern = stopwords("en")) |>
  tokens_select(min_nchar = 3)
```

Document-Feature Matrix

- DFMs are quantitative representations of the text corpus and are the basis for many text analysis methods.

Document-Feature Matrix

- DFMs are quantitative representations of the text corpus and are the basis for many text analysis methods.

```
harris_1028_dfm <- dfm(harris_1028_tokens_processed)
head(harris_1028_dfm, 10)
```

```
## Document-feature matrix of: 1 document, 710 features (0.00% sparse) and
##                    features
## docs                 can hear michelle obama good afternoon michigan we're
##   Harris_10_28.txt  13    3        3     2    3         2        6     8
##                    features
## docs                 okay
##   Harris_10_28.txt    8
## [ reached max_nfeat ... 700 more features ]
```

Word-Frequency Matrix

- *Word-frequency matrix:* Quantitative summarization of text corpus.

```r
# Summarize word frequncies.
freq_harris_dfm <- textstat_frequency(harris_dfm)
head(freq_harris_dfm, 10)
```

```
##     feature frequency rank docfreq group
## 1    people       656    1      34   all
## 2   freedom       479    2      22   all
## 3    that's       403    3      30   all
## 4    states       392    4      34   all
## 5   country       357    5      34   all
## 6    united       349    6      34   all
## 7   america       349    6      33   all
## 8   speaker       316    8      22   all
## 9  audience       304    9      17   all
## 10  believe       254   10      31   all
```

Word-Frequency Matrix

```r
freq_trump_dfm <- textstat_frequency(trump_dfm)
head(freq_trump_dfm, 10)
```

```
##     feature frequency rank docfreq group
## 1    people      4472    1      54   all
## 2   country      3166    2      54   all
## 3   they're      2805    3      53   all
## 4    that's      2510    4      53   all
## 5    you're      1522    5      53   all
## 6    really      1211    6      54   all
## 7    didn't      1140    7      53   all
## 8   america      1138    8      54   all
## 9    border      1063    9      51   all
## 10 american       928   10      54   all
```

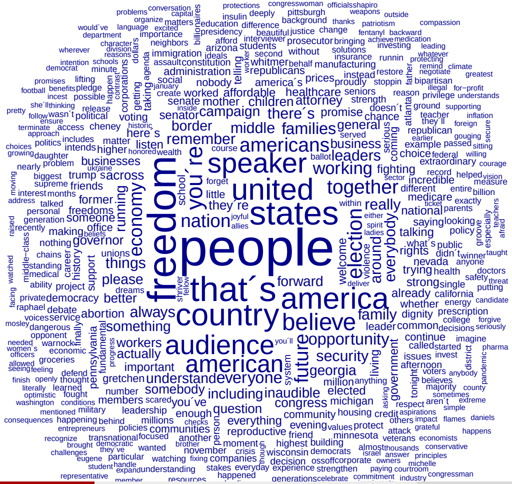
Word Cloud

- *Word cloud*: Visual representation of corpus.

Word Cloud

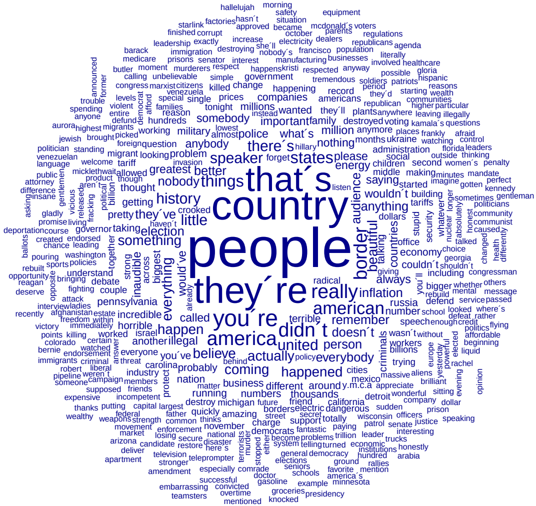
- *Word cloud*: Visual representation of corpus.

```
textplot_wordcloud(harris_dfm)
```



Word Cloud

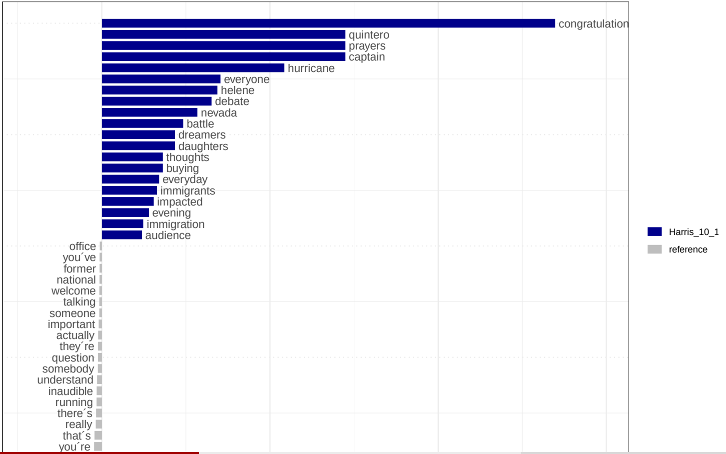
```
textplot_wordcloud(trump_dfm)
```



Keyness Plot

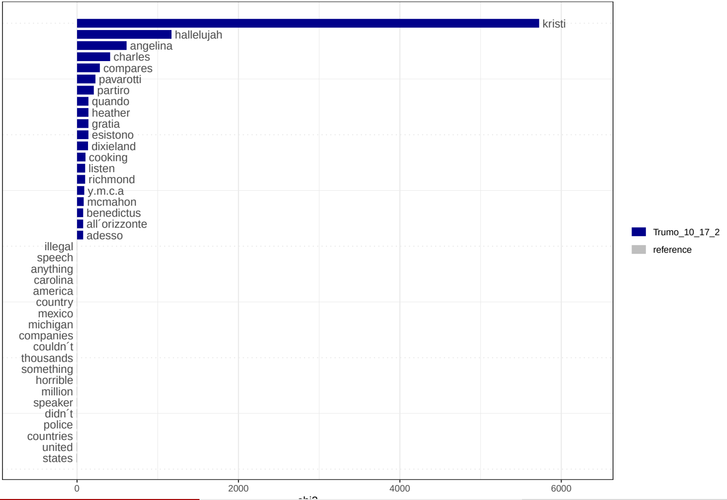
- *Word "keyness"* for specific group of documents:

```
harris_keyness <- textstat_keyness(harris_dfm)
textplot_keyness(harris_keyness)
```



Keyness Plot

```r
trump_keyness <- textstat_keyness(trump_dfm)
textplot_keyness(trump_keyness)
```



Section 2

Text Content Analysis

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy
3. Domestic policy

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy
3. Domestic policy
4. Defense

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy
3. Domestic policy
4. Defense
5. Foreign policy

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy
3. Domestic policy
4. Defense
5. Foreign policy

We could do this manually for 2024, but it would either take forever or we would need to pay RAs tons of money.

Analyzing Speech Text Data for Content

In *The Message Matters* (2009), Vavreck manually classified tons of speeches into five main categories:

1. Traits
2. Economy
3. Domestic policy
4. Defense
5. Foreign policy

We could do this manually for 2024, but it would either take forever or we would need to pay RAs tons of money.

Instead, let's try two methods of text content analysis: *structural topic modelling* and *large language model (LLM) classification*.

Structural Topic Modelling (STM)

- STM is a variation on a Bayesian technique called latent dirichlet allocation (LDA) that incorporates document-level features (metadata) into determining what topics the documents are talking about.

Structural Topic Modelling (STM)

- STM is a variation on a Bayesian technique called latent dirichlet allocation (LDA) that incorporates document-level features (metadata) into determining what topics the documents are talking about.
- It's purpose is to uncover latent topics in text and, in more advanced settings, model the relatinoships between topics and metadata like document date, author, etc.

Structural Topic Modelling in R

- `RStudio live coding!`

Many, Many Choices with Pre-Processing

Many things specific to our speeches data that may be good to pre-process to get less noisy, more interpretable results:

Many, Many Choices with Pre-Processing

Many things specific to our speeches data that may be good to pre-process to get less noisy, more interpretable results:

1. remove music lyrics

Many, Many Choices with Pre-Processing

Many things specific to our speeches data that may be good to pre-process to get less noisy, more interpretable results:

1. remove music lyrics
2. remove guest speakers (combine only text after the candidates themselves)

Many, Many Choices with Pre-Processing

Many things specific to our speeches data that may be good to pre-process to get less noisy, more interpretable results:

1. remove music lyrics
2. remove guest speakers (combine only text after the candidates themselves)
3. remove names/contractions that may be long but clutter up the text

Many, Many Choices with Pre-Processing

Many things specific to our speeches data that may be good to pre-process to get less noisy, more interpretable results:

1. remove music lyrics
2. remove guest speakers (combine only text after the candidates themselves)
3. remove names/contractions that may be long but clutter up the text

But, I ~~was too lazy to do this last night~~ may have found a more efficient way to do text analysis in the modern day using LLMs!

LLM Classification with Gemini!

Google Gemini

LLM Classification in R

- RStudio live coding!

Section 3

Course Conclusion

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions
2. ML I: OLS, LASSO, Ridge, E-net, Decision Trees, Random Forests, Ensembles, Super Learners

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions
2. ML I: OLS, LASSO, Ridge, E-net, Decision Trees, Random Forests, Ensembles, Super Learners
3. ML II: in-sample and out-of-sample error assessment, cross-validation, etc.

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions
2. ML I: OLS, LASSO, Ridge, E-net, Decision Trees, Random Forests, Ensembles, Super Learners
3. ML II: in-sample and out-of-sample error assessment, cross-validation, etc.
4. ML III: basics of quantitative textual data analysis
5. Many seminal political science theories and results about voter psychology, electoral behavior and institutions, shocks, and campaigns.

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions
2. ML I: OLS, LASSO, Ridge, E-net, Decision Trees, Random Forests, Ensembles, Super Learners
3. ML II: in-sample and out-of-sample error assessment, cross-validation, etc.
4. ML III: basics of quantitative textual data analysis
5. Many seminal political science theories and results about voter psychology, electoral behavior and institutions, shocks, and campaigns.
6. How to put skin in the game making difficult predictions, learn from our forecasts, improve our models, and use prediction and forecasting to advance social scientific knowledge about voting and elections during a fascinating and—of course—chaotic time in the U.S. and around the world.

Concepts That We Have Learned and Enjoyed

1. Principles of statistical learning and making predictions
2. ML I: OLS, LASSO, Ridge, E-net, Decision Trees, Random Forests, Ensembles, Super Learners
3. ML II: in-sample and out-of-sample error assessment, cross-validation, etc.
4. ML III: basics of quantitative textual data analysis
5. Many seminal political science theories and results about voter psychology, electoral behavior and institutions, shocks, and campaigns.
6. How to put skin in the game making difficult predictions, learn from our forecasts, improve our models, and use prediction and forecasting to advance social scientific knowledge about voting and elections during a fascinating and—of course—chaotic time in the U.S. and around the world.

Wow! That's a lot!

Thank You!!!!!!!!!

- Thank you very much for an incredible course. I have been so impressed by and proud of your work throughout the class.

Thank You!!!!!!!!!

- Thank you very much for an incredible course. I have been so impressed by and proud of your work throughout the class.
- It was an honor and a pleasure to be your TF and get to know you this semester.
- Please feel free to reach out to me about anything in the future!