Introduction to Computational Text Analysis

Florida State Summer Methods Workshop

Dr. Rochelle Terman

Department of Political Science
University of Chicago

May 2019

Instructors

- Main Instructor: Dr. Rochelle Terman (Department of Political Science, University of Chicago)

Core Learning Objectives

Ultimate Goal: Introduce students to modern computational text analysis techniques and provide an orientation for those wishing to go further with text analysis in their own research.

Core Learning Objectives

Ultimate Goal: Introduce students to modern computational text analysis techniques and provide an orientation for those wishing to go further with text analysis in their own research.

Proximate Goals

1) Learn about the main methods and techniques involved in modern computational text analysis.
2) Be able to load, preprocess, and conduct simple analysis on text data.
3) Know where to go next in their pursuit of more advanced computational text methods..

Course Outline

Topics

- Overview of Computational Text Analysis
- Preprocessing Texts
- Dictionary methods / Sentiment Analysis
- Distinctive Words
- Text similarity / distances
- K-means Clustering
- Topic Modeling

This Course Will Not

- Go into the technical details behind text analysis methods, such as optimization algorithms and theoretical properties.
- Cover all text analysis tools, or even most of them.
- Teach you how to scraping or acquiring texts.

Format of the Course

Semi flipped classroom

- 1/2 lecture, 1/2 coding in R.
- Bring your laptop, prepare to close it.
- Work with a friend, especially if you're computer isn't working.

Why Computational Text Analysis

- We care about language.

Why Computational Text Analysis

- We care about language.
- Social Scientists / Humanists have always used texts as data.

Why Computational Text Analysis

- We care about language.
- Social Scientists / Humanists have always used texts as data.
- There are costs to large-scale text analysis.

Why Computational Text Analysis

- We care about language.
- Social Scientists / Humanists have always used texts as data.
- There are costs to large-scale text analysis.
- Computers can lower these costs.

Text Analysis in Political Science

- Political speeches and deliberations ⇝ internal political workings of governments.

Text Analysis in Political Science

- Political speeches and deliberations ⇝ internal political workings of governments.
- Electoral manifestos ⇝ parties, political systems, election shifts.

Text Analysis in Political Science

- Political speeches and deliberations ⇝ internal political workings of governments.
- Electoral manifestos ⇝ parties, political systems, election shifts.
- Newspapers ⇝ media attention and political events.

Text Analysis in Political Science

- Political speeches and deliberations ⇝ internal political workings of governments.
- Electoral manifestos ⇝ parties, political systems, election shifts.
- Newspapers ⇝ media attention and political events.
- Blogs and social media ⇝ public opinion and communication.

Acquiring texts: Sources

Where to get texts:

- Online databases, e.g. LexisNexis, Comparative Manifesto Project
- Websites (Scraping, APIs)
- Archives (High-quality scanner + optical character recognition)

Acquiring texts: Sources

Where to get texts:

- Online databases, e.g. LexisNexis, Comparative Manifesto Project
- Websites (Scraping, APIs)
- Archives (High-quality scanner + optical character recognition)

Sources we'll be analyzing:

- Monographs (Machiavelli's Prince, British Fiction)
- News Articles (about women around the world)
- Song Lyrics (Taylor Swift's Catalogue)
- Press Releases (by U.S. congressperson)

Acquiring texts: Digitization

- Goal: machine readable text

Acquiring texts: Digitization

- Goal: machine readable text
- plain text (.txt or .csv) file.

Acquiring texts: Digitization

- Goal: machine readable text
- plain text (.txt or .csv) file.
- Encoded in UTF-8, ASCII

Acquiring texts: Digitization

- Goal: machine readable text
- plain text (.txt or .csv) file.
- Encoded in UTF-8, ASCII
- Metadata (author, date)

Acquiring texts: Digitization

- Goal: machine readable text
- plain text (.txt or .csv) file.
- Encoded in UTF-8, ASCII
- Metadata (author, date)
- Directory of .txt's or a “tidy” dataset

Acquiring texts: Digitization

- Goal: machine readable text
- plain text (.txt or .csv) file.
- Encoded in UTF-8, ASCII
- Metadata (author, date)
- Directory of .txt's or a “tidy” dataset
- Preprocessing to extract the most important information. (We'll cover this in-depth.)

4 Principles of Computational Text Analysis

From Grimmer and Stewart (2013):

- All Quantitative Models of Language Are Wrong – But Some Are Useful.

4 Principles of Computational Text Analysis

From Grimmer and Stewart (2013):

- All Quantitative Models of Language Are Wrong – But Some Are Useful.
- Quantitative methods for text amplify resources and augment humans.

4 Principles of Computational Text Analysis

From Grimmer and Stewart (2013):

- All Quantitative Models of Language Are Wrong – But Some Are Useful.
- Quantitative methods for text amplify resources and augment humans.
- There is no globally best method for automated text analysis.

4 Principles of Computational Text Analysis

From Grimmer and Stewart (2013):

- All Quantitative Models of Language Are Wrong – But Some Are Useful.
- Quantitative methods for text amplify resources and augment humans.
- There is no globally best method for automated text analysis.
- Validate, Validate, Validate.

An Overview of Methods

Two broad approaches to computational text analysis:

1. Supervised methods: We identify what we're interested in first, and then use computers to extend our insights to a larger population of unseen documents.

An Overview of Methods

Two broad approaches to computational text analysis:

1. Supervised methods: We identify what we're interested in first, and then use computers to extend our insights to a larger population of unseen documents.
2. Unsupervised methods: We do not specify the conceptual structure of the texts beforehand. Instead, we use the model to discover a structure that best explains the documents.

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

1) Set of known categories
    - Positive Tone, Negative Tone
    - Pro-war, Ambiguous, Anti-war

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

1) Set of known categories
    - Positive Tone, Negative Tone
    - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
    - Coding done by human coders
    - Training Set: documents we'll use to learn how to code
    - Validation Set: documents we'll use to learn how well we code

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

1) Set of known categories
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
   - Coding done by human coders
   - Training Set: documents we'll use to learn how to code
   - Validation Set: documents we'll use to learn how well we code
3) Set of unlabeled documents that we want to classify

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

1) Set of known categories
    - Positive Tone, Negative Tone
    - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
    - Coding done by human coders
    - Training Set: documents we'll use to learn how to code
    - Validation Set: documents we'll use to learn how well we code
3) Set of unlabeled documents that we want to classify
4) Method to extrapolate from hand coding to unlabeled documents (dictionary methods, logistic regression, naive bayes etc.)

Components to Supervised Learning Method

Supervised methods: Hand coding is used to train, or supervise, statistical models to classify texts in pre-determined categories.

1) Set of known categories
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
   - Coding done by human coders
   - Training Set: documents we'll use to learn how to code
   - Validation Set: documents we'll use to learn how well we code
3) Set of unlabeled documents that we want to classify
4) Method to extrapolate from hand coding to unlabeled documents (dictionary methods, logistic regression, naive bayes etc.)
5) Validate by comparing *predicted* label to actual (hand-coded) *label.*

Components to Unsupervised Learning Methods

Unsupervised methods: Discover new ways of organizing texts that are theoretically useful, but perhaps understudied or previously unknown.

Components to Unsupervised Learning Methods

Unsupervised methods: Discover new ways of organizing texts that are theoretically useful, but perhaps understudied or previously unknown.

1) Set of unlabeled documents that we want to classify

Components to Unsupervised Learning Methods

Unsupervised methods: Discover new ways of organizing texts that are theoretically useful, but perhaps understudied or previously unknown.

1) Set of unlabeled documents that we want to classify
2) Method to discover categories and then classify documents into those categories (k-means clustering, topic models)

Components to Unsupervised Learning Methods

Unsupervised methods: Discover new ways of organizing texts that are theoretically useful, but perhaps understudied or previously unknown.

1) Set of unlabeled documents that we want to classify
2) Method to discover categories and then classify documents into those categories (k-means clustering, topic models)
3) Interpretation skills to assign labels to categories and understand what they mean

Methods we'll be covering

- Preprocessing
- Dictionary methods / sentiment analysis (Supervised)
- Distinctive words
- Text as geometry (similarity and distance)
- K-means Clustering
- Topic modelling (Unsupervised).

Methods we'll be covering

- Preprocessing
- Dictionary methods / sentiment analysis (Supervised)
- Distinctive words
- Text as geometry (similarity and distance)
- K-means Clustering
- Topic modelling (Unsupervised).

Methods we won't be covering

- Text scaling
- Complex supervised methods
- Information retrieval
- Natural Language Processing

Let's Get Started!

1. Download the Class Repo as a zip file: https://github.com/rochelleterman/FSUtext
2. Unzip the file in a location of your choice.
3. Find the path of the repo and write it down.
4. Download the R packages listed in B-Tech-Requirements.md.