This syllabus will be updated: http://faculty.washington.edu/jwilker/559/2018/559Syllabus2018.pdf
Slides here

POLS 559: Text as Data

Winter 2018

John Wilkerson (jwilker@uw.edu)

Class meets TH 1:30=4:20, Savery 164

Office Hours: M (1-3 or by apt) Smith 221C

This class introduces computational approaches for collecting, preparing and analyzing text as data. It is not a programming class but you will need to do some programming to complete assignments. The main goal of the class is to survey computational approaches that have essentially the same objectives as other quantitative studies – counting, scaling and grouping. Like any quantitative project, validity – are we really capturing what we are trying to measure? – is of central concern.

We begin by considering the goals and practice of non-computational content analysis. We then work through the stages of a typical text as data project - getting text, converting it to data, and analysis. The analysis component covers several approaches commonly used in the social sciences.

Finally, the main benefit of computational methods is the ability to scale up an analysis. Each participant designs and executes an original and ambitious project using the methods covered in the course.

Resources

There are no required books for this class. A lot of books on data science are being published lately and they are worth reading. However, none are closely related to how this class is taught (there is no book about Quanteda for example). Many of these are also available on-line.

The Coding Manual for Qualitative Researchers, by Saldana (we'll read a chapter but this is a very helpful book)
R for Data Science, by Wickham and Grolemund (Wickham is a leading R developer)
Data Vizualization for Social Science, by Healy (how to present results using R)
Natural Language Processing with Python, by Bird, Klein, Loper
Learning Python, by Lutz *(If you are really interested in the Python programming language, this is the intro.)*

Arguably the most important development in terms of learning programming languages is the internet. Most of the answers to your questions can be found by using Google search. There are also many helpful on-line tools, such as:

- Learn Python in 10 Minutes! (https://www.stavros.io/tutorials/python/)
- The Python Tutorial (https://docs.python.org/2/tutorial/)
- Pythex (http://pythex.org/) for testing python regular expressions

Grading

- **Participation (20%) –** Class attendance and contributions to in class discussions and activities. Readings are to be completed by the listed date.
- **Homeworks (30%) –** They are due **Tues** evening on Catalyst unless otherwise noted.
- **Research Project (50%)** . This is where you demonstrate what you have learned. We will be talking about potential projects right away. My office door is open! **Proposal** (Feb 14); **Draft** (Mar 5); **Final project** (Mar 15)

(See glossary of text as data terms at the end of the syllabus)

January 4 – Introduction and Installation

Readings (to be completed before class)

- Language from police body camera footage shows racial disparities in officer respect. Voigt et al
- Tracing the Flow of Ideas in Legislatures: A Text Reuse Approach. Wilkerson et al
- Crowd-Sourced Text Analysis: Reproducible and Agile Production of Political Data. Benoit et al

Activity (try to make some progress before class)

- Install the Python 2.7 version of Anaconda and create an icon for Spyder or Jupyter (similar to R Studio) on your laptop.
    - Use shift/enter to 'run' a command in Spyder
    - Do a little addition to confirm that Python is working
- Install the most recent version of R (3.4.) and R Studio (or some editor)
    - Install the quanteda and ggplot2 packages
    - Use control/enter to 'run' a command in RStudio
    - Do a little addition
- If things are going well, try these Python challenges! http://faculty.washington.edu/jwilker/559/Python%20challenges.pdf

Homework (doesn't need to be turned in)

- If you are not very familiar with R, please do the introductory lessons in SWIRL
- If you already know R, how about a little Python?

It is essential that everyone have R and Python working before class on Jan 11. If you are having issues and CSSCR consulting can't help, please let me know well ahead of class.

January 11 – The big picture. Steps in a text as data project. R and Python basics

Readings (to be completed before class)

- Large scale computerized text analysis in political science: Opportunities and challenges Casas and Wilkerson
- Text as Data: The Promises and Pitfalls of Automated Content Analysis. Grimmer and Stewart

Activity: Python v R: Please read these in advance

- Python for R Users" "Python for R – Strings" "Python for R – Dicts and Tuples"
- A whole lotta Python examples
- Regular Expressions are the Bomb RegularExpressions.py

Homework 1 (this a quanteda exercise to be submitted by Tuesday night)

January 18 – Stepping Back: *Content Analysis: Objectives, Process and Evaluation*

Saldana considers what researchers want from text and the diversity of approaches that might be pursued. He then turns to the different things one might want to capture from texts. Finally, he describes how researchers go about developing coding systems. Baumgartner et al write about their experiences of developing a coding system that is now widely used (comparativeagendas.net).

Readings (to be completed before class)

- Saldana, Chapter 3 (big file, be patient!)
- *Lessons from the Trenches Baumgartner, Jones, McLeod*
- Reliability and Validity in Research *Roberts et al. [difference between R and V? Types of V]*
- Reliability in Content Analysis, *Krippendorf [Which measure? Four recommendations?]*
- "The Unreliability of Measures of Intercoder Reliability, and What to do About it" Grimmer et al. [conflating reliability and vaidity]

Activity: Getting text

- Converting pdfs to text NewVersion
- A whole lotta Python examples includes examples of downloading documents, webpages and using APIs

Homework 2 (to be submitted by Tuesday night)

January 25 – *Unsupervised Machine Learning*

Readings (to be completed before class)

- Domingues, The Master Algorithm, Chapters 2, 3 (sorry about the format)

Unsupervised machine learning

- Mining the Dispatch
- Probabilistic Topic Models, Blei
- Structural Topic Models for Open-ended Survey Responses Roberts et al see also STM Vignette
- Reading the tea leaves Chang, Graber, Blei (*beware of objective fit measures like perplexity*)

Activity: Topic model selection and interpretation

- LDA topic model.R GenSimLDAExample.py

Homework3: (to be submitted by Tuesday Jan 6)

February 1 - *Supervised Machine Learning*

Readings (to be completed before class)

- Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods. Collingwood and Wilkerson
- Thumbs up: Sentiment Classification using Machine Learning Techniques. Bo et al
  See also page 27 of this book for a discussion of unsupervised approaches to sentiment analysis
- A Method of Automated Non-Parametric Content Analysis for Social Science" Hopkins and King
- How the Chinese Government Fabricate Social Media Posts King, Pan and Roberts
- Legislative hitchhikers, Casas et al
- Machine Bias

Activity: Projects discussion

Homework4: (to be submitted by Tuesday Jan 6)

February 8 – *Meanings matter: Natural Language Pt 1*

Gender and Teacher Reviews (Bookworm)
Appendix for body cam study (from first week of class)
Event Data Analysis, Schrodt and Gerner
Clause Analysis *Van Atteveldt et al.*
Distributed Representations of Words and Phrases *Mikolov et al.*

TopicSimilarity termslist cosine R script

NLTK Natural language Processing examples

Homework 5 – Sentiment bakeoff! Sentiment.py Some Sentiment Lexicons
(Research proposal due Feb 14)

February 15 – *Natural Language: Pt 2: Vector representation, text reuse and vocal pitch*

Word2vec Introduction // myword2vecintro load.py word2vec.py
Political Ideology Detection Using Recursive Neural Networks Iyyer et al Blog Summary

Infectious Texts: Modeling Text Reuse in Nineteenth-Century Newspapers," Smith Cordell, Dillon ViralTexts
textreuse.py

Emotional Arousal Predicts Voting on the U.S. Supreme Court, Dietrich, Enos and Sen
Replication files

Homework 6

February 22 – No class

March 1 – Deep learning and Computer vision

ImageNet
Amazon Rekognition
PyTorch

This cat sensed death: what if computers could too?, NYT
Facial Recognition is Accurate: If You're a White Guy

Images as Data: Computer Vision for Social Science Research, Casas and Webb Williams
Introduction to deep learning and computer vision, Casas

Computer vision module (see instructions about confirming that you have a bash terminal on your computer; you won't actually do this until class but we don't want people trying to install bash at that time) vpgtools (dependencies)

Of possible interest (yikes!): Mastering the game of Go without human knowledge, Silver et al

---

March 8 – Setting up an AWS Instance

http://faculty.washington.edu/jwilker/559/2018/SettingupanAWSInstance.docx

Some Terms of Interest

Precision – proportion of predicted X cases that are true Xs. (precision errors are false positives)
Recall – proportion of true Xs that are predicted Xs (recall errors are false negatives)
F-Score – the harmonic mean of precision and recall
Validity – on average does it accurately captures the concept being measured?
Reliability –does it produce the same result each time
Bias – A measure can be reliably invalid
Confusion matrix – crosstabulation of *actua*l versus *predicted* results that is used to study and learn from the distribution of errors.
Annotate = classify = label = code ("code' is discouraged because it also refers to programming)
Token – any element of a document (e.g. a word; space; semicolon).
Tokenization (aka text segmentation) - the process of breaking up a text into meaningful elements (e.g. spaces can be used to distinguish words as tokens)
Feature – any token that is relevant to the text task.
Parsing – Generally, the process of systematically partitioning a text into meaningful components (such as sentences or words). In NLP, a formal methodology for tagging words according to linguistic rules (see *Stanford Parser*).
Normalization – eliminating differences in punctuation, such as removing capitalization
Stemming – shortening words to their stem, base or root form (e.g. fishing–fish)
Stopword – words that are not considered to be valuable features and are therefore excluded (e.g. the)
Regular expression – a search pattern (for example to find any date in a text)
Disambiguation – NLP process of identifying different ways of referring to the same entity. For example, blogs might variously refer to President Obama as 'Barack,' ' Obama,' 'The One,' 'the President' etc.
Named entity – NLP process of classifying words into predefined categories (e.g. person names, organizations, locations, subjects, percentages, etc).
Semantic – Broadly, the meaning of a word.
Sentiment – positive-negative polarity in classification (e.g. from hate to love, liberal to conservative, etc).
Algorithm – a mathematical set of instructions about how to convert a set of inputs to an output.
Machine learning – a computer science catch-all term for statistical modeling. That said, the models tend to be different and are evaluated differently from the models covered in more conventional statistics courses.
Bag of Words (BoW) – common analytic approach that treats words as features in isolation (as opposed to NLP approaches that consider the meaning of or relationships between words)
Natural Language Processing (NLP) – a wide range of text analysis approaches informed by how people actually use words.