Computational text analysis: A practical overview

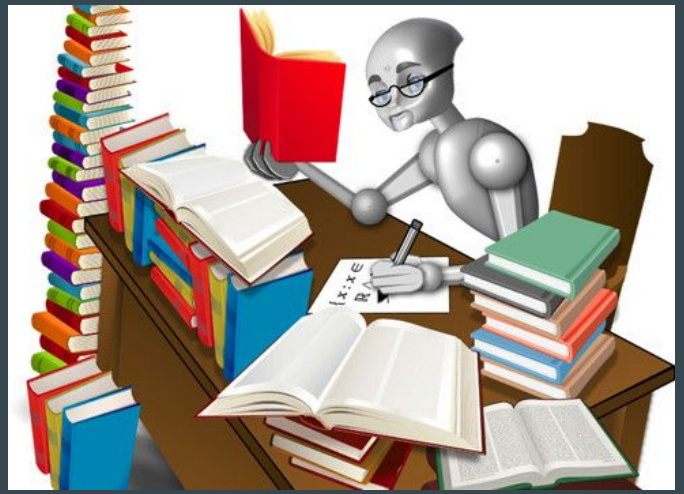
February-March 2018

Geoff Bacon

With special thanks to Ben Gebre-Medhin and Laura Nelson

Computational text analysis (CTA)

- Making computers analyze texts for us
- Closely related to natural language processing (NLP) and computational linguistics



Goals of this workshop series

- Provide a general roadmap of CTA possibilities
- Help you understand where to invest your time
- Build intuitions about different methods
- Understand at a high-level
  - how each method works
  - their strengths and weaknesses
  - what kinds of questions they answer
- Practice implementing methods
- Know where to go from here for your project

Overview of this workshop series

- Four 2-hour sessions
- Same time and place each week
- No programming experience is required
- But it will be more useful to you if you do
- Day 1: Introduction and system set-up
- Day 2: Preprocessing
- Day 3: Unsupervised methods
- Day 4: Supervised methods

Introductions

- Instructor
- Participants
  - Name
  - Affiliation on campus
  - Any immediate projects or uses cases?
  - Any familiarity with Python or R?

- If you do have immediate projects, bring them in!

Types of languages

- Natural languages

Time flies like an arrow. Fruit flies like a banana.

- Artificial languages

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_n X_n + \epsilon$$

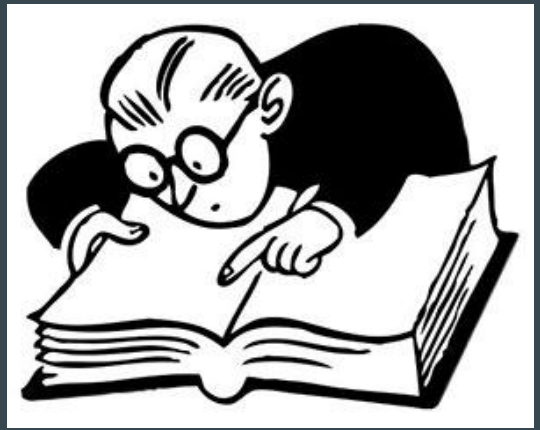
```
import scipy
from scipy import sparse

n = 200000
matrix = scipy.sparse.rand(n, n, density=.001)
print(matrix)
```

How do humans analyze texts?

We need to steer clear of this poverty of ambition, where people want to drive fancy cars and wear nice clothes and live in nice apartments but don't want to work hard to accomplish these things. Everyone should try to realize their full potential.

- Barack Obama

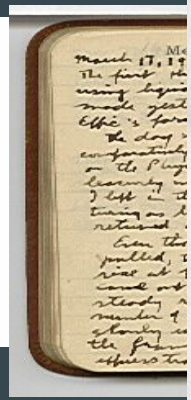
Close reading



The promise of distant reading

- Scale/speed
- Reproducibility
- Does not have human biases
- Has other (at times unknown) biases
- Consistent

Text as data




FIELD NOTES
48-Page Memo Book
Durable Materials / Made in the U.S.A.

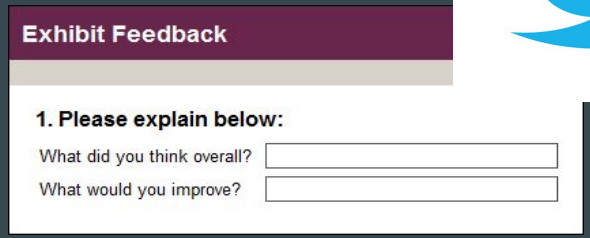
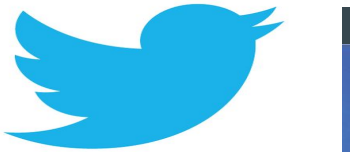


Exhibit Feedback
1. Please explain below:
What did you think overall?
What would you improve?



What can you do with CTA?

- Estimate political ideology from Twitter
- Uncover government censorship
- Relate the stock market to sentiment in the media
- Study why particular papers get cited
- Detect impending disease epidemics
- Determine who actually wrote something
- . . .

CTA lifecycle

…

CTA lifecycle

- Cooking analogy

Research question

- Domain specific
- Possible answer is encoded in text

- E.g. *What are the early warning symptoms of depression?*
- *How did different European nations react to the election of Trump?*
- *Do Twitter users react differently to mass shootings based on the ethnicity of the perpetrator?*
- *What distinguishes different styles of hip-hop?*
- *Have any of these essays been plagiarized?*

Collect data

What do we want?

- Machine readable
- Plain text = just the readable characters (i.e. letters, punctuation) plus new line.
- Not Word or pdf
- Sometimes useful to have some structure: csv, xml, html, json
- If many files, we'd like some structure (subdirectories or useful file names)

How do we get it?

- You (your collaborator or a stranger on the Internet) already have it*
- Web scraping
- API
- OCR

- *Still important to know how the data was collected

Preprocessing

What is preprocessing?

- Tokenization = separating running text into words
- Sentence segmentation/tokenization = separating words into sentences
- Text normalization = dealing with upper/lower case, spelling mistakes, removing special characters, replacing URLs, numbers, etc.
- Remove “stop words”
- Stemming/lemmatization = removing morphological affixes
- POS tagging = assigning a part-of-speech category to each word
- Syntactic parsing = assigning a (normally graph or tree) structure to a sentence
- Chunking = shallow version of syntactic parsing
- Named entity recognition = identifying the proper nouns in a text
- . . .

Preprocessing

Why do we do preprocessing?

- Because later methods require preprocessed data as input
  - Counting words requires having already identified the words of a text
  - Knowing the POS of a word might help us in knowing whether it is important (modal *can* vs noun *can*)
- Because we gain intuition about our data
  - It forces us to look at the data
  - Often coupled with exploratory data analysis (EDA)
  - We might find out that all the reviews are exactly 500 characters long, which suggests some have been truncated.

Modeling

Modeling in this workshop

- In this workshop, we'll focus on topic modeling and text classification.
- Topic modeling: input = many different texts, output = what each text is about
  - Newspaper articles
  - Emails
- Classification: input = many different texts and hand-labeled categories, output = something that can take in unlabeled texts and predict the category.
  - Hand label a bunch of documents, train a computer to mimic your hand coding
  - Spam / ham
  - positive/negative reviews
- The big difference is the need for labeled data (unsupervised vs supervised)
- Text = document, could be a review, newspaper article, journal article, whole book, tweet, . . .

Clarifying some terminology

- Text = document, a single unit under study
- Corpus = a collection of documents
- Lexical ≃ fancy name for word
- N-gram: sequences of adjacent words of length *N*
  - *poverty*: uni-gram
  - *poverty of*: bi-gram
  - *poverty of ambition*: tri-gram
- Bag of words

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Slide from Jurafsky & Manning's online course on NLP

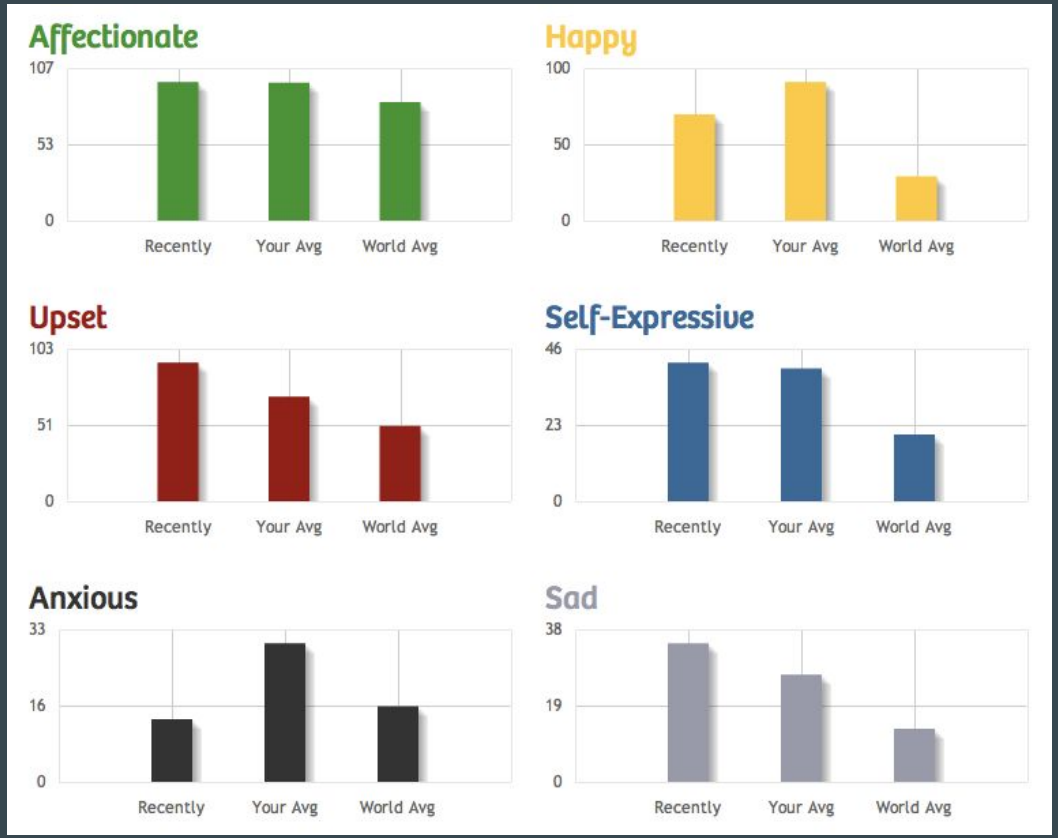
Modeling

But there's more out there

- Dictionary methods
  - Lists of positive/negative words
- Document-Term Matrix (DTM)
  - Words are rows, documents are columns, entries are number of times word appears in document.
- Term Frequency-Inverse Document Frequency (TF-IDF)
  - Modification of DTM
  - Entries are scaled by how common a word is across the whole corpus
- Clustering
  - Of DTM or TF-IDF

Modeling

Dictionary method



Now what?

- Use the insight from the preprocessing and modeling stage to understand the initial question
- In classification, this could be looking at words with high coefficients.
  - E.g. *People with depressive symptoms are more likely to talk about abstract concepts and use more negation.*
- In topic modeling, this could be qualitative inspection of the topics.
  - E.g. *In Germany the main themes centered around taxes and immigration, while in France people were more concerned about racism.*

CTA lifecycle

Questions?

Yeah, but how do I do it?

- Yourself
  - This workshop
  - Other resources (in a few slides)
  - NLTK, SpaCy, gensim, polyglot, TextBlob, etc.
- Using third-party APIs
  - Google, Amazon, Microsoft and IBM
- Work with someone
  - Fellow researchers/staff
  - D-Lab consulting

Where to go from here?

…

Further resources

- D-Lab: http://dlab.berkeley.edu/
  - Regular expressions in Python
  - NLP with nltk/SpaCy
  - Consulting
- CTAWG: http://dlabctawg.github.io/
- Lectures from Stanford's NLP class
- Jurafsky and Martin textbook

Preparing for the technical workshops

...

Anaconda

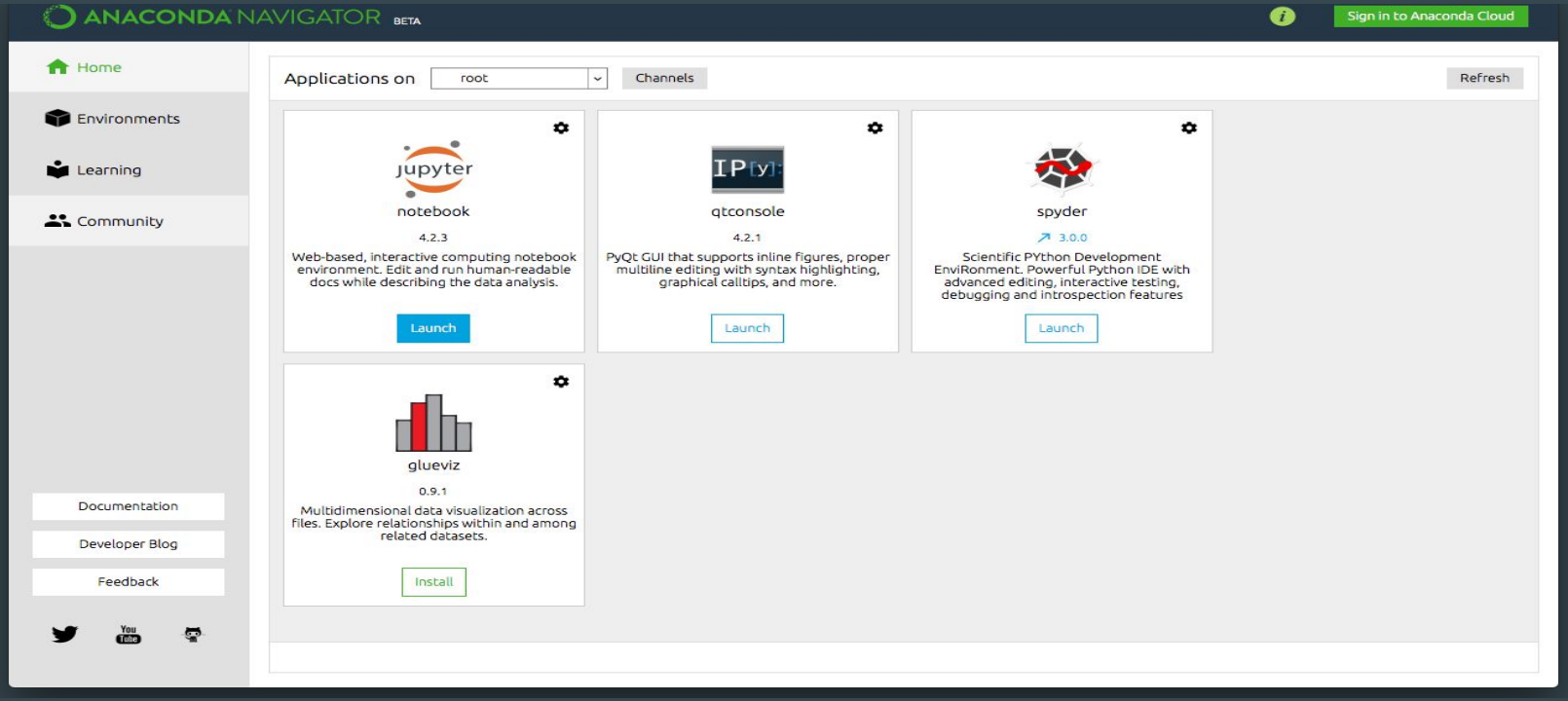
- Download and install Anaconda: https://www.anaconda.com/download/
- Anaconda is designed to make installation easy, by bundling together common data science libraries in one install
- It comes in two versions, one for Python 3 and one for Python 2.
- PLEASE DOWNLOAD THE PYTHON 3 VERSION

Please download the Python 3 version

Download Anaconda

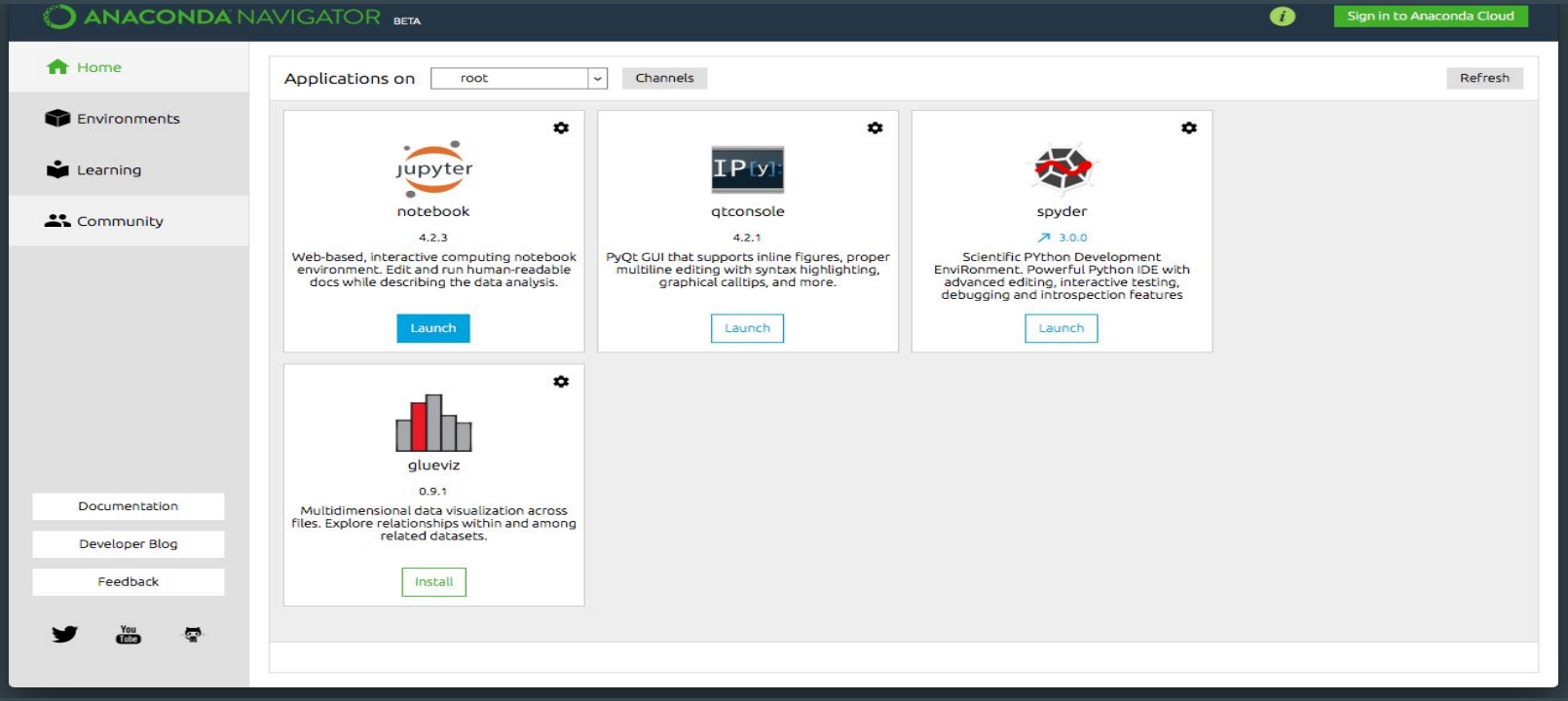
Install Anaconda

Run Anaconda

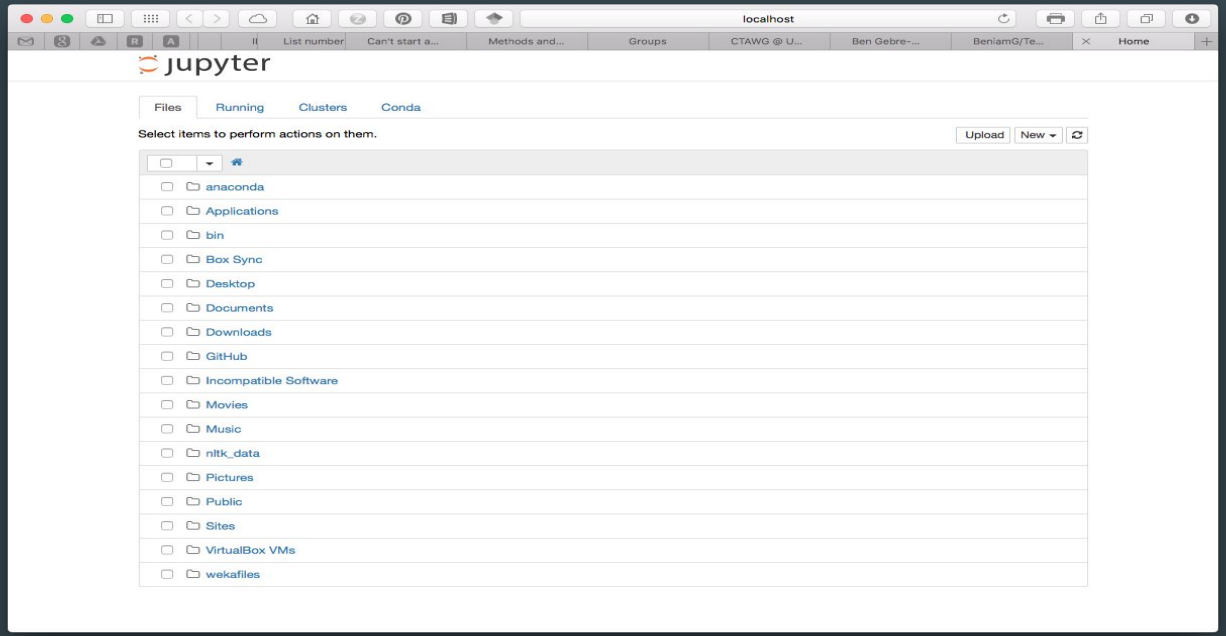


Download GitHub repository

Run Jupyter


ANACONDA NAVIGATOR BETA
Sign in to Anaconda Cloud
Home
Environments
Learning
Community
Documentation
Developer Blog
Feedback
Applications on
root
Channels
Refresh
jupyter
notebook
4.2.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
IP[y]:
qtconsole
4.2.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
spyder
3.0.0
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
glueviz
0.9.1
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Run “Intro to Python.ipynb”



Should look like this