Module 4.2: Performing Text Analysis

Basic Approaches with Python
In this module we’ll…

- Review text analysis strategies for advanced researchers
  → Make skill-appropriate recommendations
- Explore text analysis methods in-depth
  → Understand the kinds of research available in the field
- Use an HTRC dataset to conduct exploratory data analysis
  → Practice programming skills for data-driven research
- See how Sam analyzed his Creativity Corpus
  → Learn how a researcher used text analysis methods
Where we’ll end up

Use a Python library to create a list of top adjectives in a volume and a graph that visualizes word count across a volume.
Key approaches to text analysis

Broad Area: Natural Language Processing (NLP)

Using computers to understand the meaning, relationships, and semantics within human-language text

• Specific Methods:
  • **Named entity extraction**: what names of people, places, and organizations are in the text?
  • **Sentiment analysis**: what emotions are present in the text?
  • **Stylometry**: what can we learn from measuring features of style?
Key approaches to text analysis

**Broad Area: Machine Learning**

Training computers to recognize patterns.

- **Specific Methods**
  - **Topic modeling** – What thematic topics are present in the text?
  - **Naïve Bayes classification** – Which of the categories that I have named does the text belong to?
Activity: Identify the method

<table>
<thead>
<tr>
<th>Broad area</th>
<th>Specific method</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Rowling and &quot;Galbraith&quot;': an authorial analysis</td>
<td></td>
</tr>
<tr>
<td>Significant Themes in 19th Century Literature</td>
<td></td>
</tr>
<tr>
<td>The Emergence of Literary Diction</td>
<td></td>
</tr>
</tbody>
</table>

Note: Broad areas/specific methods are those defined in the previous two slides

Link to project summaries: [http://go.illinois.edu/ddrf-research-examples](http://go.illinois.edu/ddrf-research-examples)
Text analysis workflow

RAW TEXT
▼
TEXT PREPARATION
▼
TRANSLATION INTO FEATURES
▼
ALGORITHMIC USE
Features in the HTRC

- HTRC Extracted Features dataset
- Downloadable
- Structured data consisting of features
- 5 billion pages, in 13.6 million volumes

https://analytics.hathitrust.org/datasets#ef
HTRC Extracted Features (EF)

- The features are
  - Selected data and metadata
  - Extracted from raw text
- Position the researcher to begin analysis
  - Some of the preprocessing is already done
- Form of non-consumptive access
Per-volume features

- Pulled from bibliographic metadata
- Title
- Author
- Language
- Identifiers

```json
1 {  
   "id":"uc1.b3419888",  
   "metadata":{  
      "schemaVersion":"1.2",  
      "dateCreated":"2015-02-12T13:30",  
      "title":"Zoonomia = or The laws of organic life / by Erasmus Darwin. ",  
      "pubDate":"1809",  
      "language":"eng",  
      "htBibUrl":"http://catalog.hathitrust.org/api/volumes/full/htid/uc1.b3419888.json",  
      "handleUrl":"http://hdl.handle.net/2027/uc1.b3419888",  
      "oclc":"3679915",  
      "imprint":"Thomas and Andrews, 1809."
   },  
   "features":{  
      "schemaVersion":"2.0",  
      "dateCreated":"2015-02-20T23:58",  
      "pageCount":616,  
      "pages":[]
   }
}
```
Per-page features

- Page sequence
- Computationally-inferred metadata
  - Word, line, and sentence counts
  - Empty line count
  - Language

```json
{  
  "seq" : "00000035",  
  "tokenCount" : 507,  
  "lineCount" : 44,  
  "emptyLineCount" : 0,  
  "sentenceCount" : 14,  
  "languages" : [  
    {  
      "en" : "1.00"  
    }  
  ]
}
```
Public papers of the presidents of the United States (Gerald R. Ford, book 2)
Page section features

<table>
<thead>
<tr>
<th>LIST OF ITEMS</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>CABINET</td>
<td>lxvi</td>
</tr>
</tbody>
</table>

PUBLIC PAPERS OF GERALD R. FORD, JULY 21—DECEMBER 31, 1975

Appendix A—Additional White House Releases  2021
Appendix B—Presidential Documents Published in the Federal Register  2049
Appendix C—Presidential Reports to the 94th Congress, 1st Session  2057
Appendix D—Rules Governing This Publication  2061
INDEX  A–I

Public papers of the presidents of the United States
(Gerald R. Ford, book 2)
Page section features

Header, body, footer

- Line, empty line, and sentence count
- Counts of beginning- and end-line characters
- Token counts
  - Homonyms counted separately
  - Part-of-speech codes are from the Penn Tree Bank

```json
"body":{
  "tokenCount":504,
  "lineCount":43,"
  "emptyLineCount":0,"
  "sentenceCount":12,
  "tokenPosCount":{
    "synthesifis":{"NNP":1},
    "Laws":{"NNP":1},
    "beautiful":{"JJ":1},
    "philosopher":{"NN":1},
    "uponthe":{"IN":1},
    "for":{"IN":1},
```

Using HTRC Extracted Features

- Identify parts of a book
  - From descriptive metadata
- Perform any method that works with bags-of-words
  - Topic modeling
  - Dunning’s log-likelihood
- Classify volumes
  - Compare with bibliographic metadata
Do-it-yourself text analysis

- Some researchers won’t use off-the-shelf tools
  - Want more control over processes
  - Set their own parameters

- Mix-and-match approaches to create “toolkit” of strategies
The toolkit

- Researcher-dependent
- Requires understanding of statistics
- Often draws on expert collaborators
- Consists of command line tools and programming languages
Command-line tools

- Tools you download and run from the command line
  - MALLET (Java-based)
    - Topic modeling and classification
  - Stanford NLP (Java-based)
    - Natural language processing
Code libraries

- Part of programming is re-using code, often shared as libraries
- The existing code is installed via packages
- Packages are collections of bits of code, made up of modules
- Modules facilitate programming tasks
Using libraries in practice

BenM
@benmiller314

me on python: spend 2 hrs working out how to do X. discover library designed to do X better. spend 3 hrs wading into that library. repeat.

Likes
5

12:54 PM - 20 Jun 2017
Installing packages for Python

- Install using a package manager
- Example: pip
  - Package manager for Python
  - Generally comes with Python install
- Basic syntax
  - pip install <name>
- Other installers: Homebrew, Conda
Using a library’s module in a script

- Once library is installed…
- Must declare that it will be used at the top of the script
  - “Import” the module
- Use the desired module in the body of the script

```
from htnc_features import FeatureReader
import pandas as pd
import sys

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print("Missing filename argument")
        sys.exit(1)

filename = sys.argv[1]
idx = pd.IndexSlice

def get_proper_nouns(vol):
    tl = vol.tokenlist(pages=False)
    tl.index = tl.index.droplevel(0)
    try:
        proper_nouns = tl.loc[idx[;,:,'VB','VBD','VBG','VBN','VBP','VBZ'],]
        proper_nouns.index = proper_nouns.index.droplevel(2)
        return proper_nouns[proper_nouns['count'] < 150]
    except:
        return pd.DataFrame()
```
Python text analysis libraries

- Machine learning
  - SciKit Learn

- Data science
  - Pandas

- Natural language processing
  - Natural Language Toolkit (NLTK)

Example: `nltk.words_tokenize()`
  - Can you guess what this does?
HTRC Feature Reader Python library

- Python library for working with HTRC Extracted Features
  - Bits of pre-written code to parse the JSON
- Install using a package manager, like pip
  - Source code lives on Github
- Also need to have Pandas installed
  - Remember: pandas = Python library for working with data
  - Luckily it’s included in PythonAnywhere!
Sample Reference Question

I’m a student in history who would like to incorporate digital methods into my research. I study American politics, and in particular I’d like to examine how concepts such as liberty change over time.

Approach: view adjectives in Extracted Features file
Hands-on activity

In this activity, you will install the Feature Reader library and run a Python script to create a list of the most-used adjectives and the number of times they occur in different sets of volumes of presidential papers (using the Extracted Features files of these volumes).

What You Need:

Script: top_adjectives.py

Files for analysis, in 2 directories:

• 1970
• 1930
Files and directories

- 1930 and 1970 directories and required files should already be in your files in PythonAnywhere

- Check using the “Browse files” button

- 1970/ = Extracted Features files of presidential speeches from the 1970s (corresponds to our political science workset!)

- 1930/ = Extracted Features files of presidential speeches from the 1930s
Install Feature Reader library

Bash console 4179031

19:56 ~ $ pip install --user htrc-feature-reader

```
pip install --user htrc-feature-reader
```
Examine code

```python
from htrc_features import FeatureReader
import glob
import pandas as pd
import sys

decade = sys.argv[1]
paths = glob.glob(decade+'/*.json.bz2')
idx = pd.IndexSlice
fr = FeatureReader(paths)
vol = next(fr.volumes())
_tl = vol.tokenlist(pages=False)
_tl.index = tl.index.droplevel(0)
adjectives = tl.loc[idx[:,('JJ')]]
adj_dfs = [adjectives for vol in fr.volumes()]
all_adj = pd.concat(adj_dfs).groupby(level='token').sum().sort('count', ascending=False)[:50]
print(all_adj)
```

Prepares us to use the libraries needed for this task.

‘JJ’ is the code for adjectives from the Penn Treebank.
Open shell and run command

```
13:45 ~ $ python top_adjectives.py 1970
```
View results

<table>
<thead>
<tr>
<th>token</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>21248</td>
</tr>
<tr>
<td>_</td>
<td>17824</td>
</tr>
<tr>
<td>other</td>
<td>11552</td>
</tr>
<tr>
<td>own</td>
<td>10400</td>
</tr>
<tr>
<td>new</td>
<td>9984</td>
</tr>
<tr>
<td>good</td>
<td>9440</td>
</tr>
<tr>
<td>great</td>
<td>3920</td>
</tr>
<tr>
<td>American</td>
<td>7936</td>
</tr>
<tr>
<td>many</td>
<td>7840</td>
</tr>
<tr>
<td>major</td>
<td>6480</td>
</tr>
<tr>
<td>last</td>
<td>6288</td>
</tr>
<tr>
<td>public</td>
<td>6224</td>
</tr>
<tr>
<td>important</td>
<td>6096</td>
</tr>
<tr>
<td>first</td>
<td>5808</td>
</tr>
<tr>
<td>such</td>
<td>5760</td>
</tr>
<tr>
<td>economic</td>
<td>5504</td>
</tr>
<tr>
<td>human</td>
<td>5424</td>
</tr>
<tr>
<td>international</td>
<td>5216</td>
</tr>
<tr>
<td>national</td>
<td>4864</td>
</tr>
<tr>
<td>same</td>
<td>4688</td>
</tr>
<tr>
<td>next</td>
<td>3920</td>
</tr>
<tr>
<td>nuclear</td>
<td>3840</td>
</tr>
<tr>
<td>local</td>
<td>3744</td>
</tr>
<tr>
<td>foreign</td>
<td>3696</td>
</tr>
<tr>
<td>political</td>
<td>3648</td>
</tr>
<tr>
<td>comprehensive</td>
<td>3488</td>
</tr>
<tr>
<td>few</td>
<td>3472</td>
</tr>
<tr>
<td>sure</td>
<td>3440</td>
</tr>
<tr>
<td>possible</td>
<td>3408</td>
</tr>
</tbody>
</table>
On your own

- Compare the adjectives used by presidents in the 1970s with those used in the 1930s.
- Work and discuss with your neighbor.

- How do you need to change your command?
- What differences do you see? Similarities?

Challenge

Try modifying the script to search verbs or another part of speech. (Hint: Don’t forget the Penn Tree Bank!)
Exploratory data analysis

- Approach for getting familiar with data
- Easier to recognize patterns (and problems)
  - Hard to see trends in a spreadsheet or text file!
- There are whole books about it

Strategies:
- Plot raw data
- Plot simple statistics
- Compare plots to look for patterns
Visualization libraries

- **Python**
  - `Matplotlib` library, `pyplot()` function
  - `ggplot` library

- **Others:**
  - R: `ggplot2`
  - D3.js: JavaScript library for visualizations
I’m a student in history who would like to incorporate digital methods into my research. I study American politics, and in particular I’d like to examine how concepts such as liberty change over time.

Approach:

View word count over a volume using an Extracted Features file
Hands-on activity

- In this activity, you will create a visualization using the native plotting functionality in Pandas, called pyplot, to see the word count over a volume based on its Extracted Features file.

Script: word_count.py

Files for analysis: mdp.49015002203033.json.bz2 (Presidential Papers of the Presidents of the United States: Gerald Ford)
Examine code

```python
from htrc_features import FeatureReader
import os
import glob
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
matplotlib.use("Agg")

path = glob.glob('1970/mcp.49015002203033.json.bz2')
fr = FeatureReader(path)
vol = fr.first()
tokens = vol.tokens_per_page()
tokens.plot()
plt.savefig("words.png")
```

- Imports our modules so we can call functions later, including matplotlib's pyplot.
- Set variable "tokens".
- Plot tokens and save plot as png.
Open shell and run command

```
12:28 ~ $ python word_count.py
```

```
python word_count.py
```
The graph visualizes how many words appear on each page in the volume.

Page numbers are on the horizontal axis, and the word count is on the vertical axis.

E.g. The 600th page has about 350 words.
On your own

- Can you modify the script to look at another volume?
- Edit, save, and run the file
  - Hint: Change the path to another volume in the script
- View and discuss with your neighbor

☞ See Handout p. 3
Another look

- Even basic counts can lead to sophisticated research
- Ted used HTRC EF
- Classified genre at the page level

From paper - graph of diction patterns between genres, using frequency counts (Underwood and Sellers, 2012)
Case Study: *Inside the Creativity Boom*

After reducing Creativity Corpus to pages containing forms of creativ*: 

- Performed topic modeling on those pages
- Ended up with topics that reflect what themes are prevalent around concept of “creativity” in the 20th century
- Graphed the topics over time to see how their usage changed
Case Study: *Inside the Creativity Boom*

- Topics that decreased in usage over time
  - god, christ, jesus, creation, word
  - species, animals, natural, plants, soil
  - nature, mind, creative, world, human
  - invention, power, creative, own, ideas

[Graph showing creativity topics with falling usage]
Case Study: *Inside the Creativity Boom*

- Topics that increased in usage over time
  - advertising, media, marketing, sales, television
  - economic, development, capital, economy, production
  - poetry, language, poet, poets, poems
  - social, creative, study, development, behavior

*Creativity topics with increasing usage*
Discussion

- *In what ways can librarians support advanced text analysis research?*

- *What additional skills would you need to learn in order to do so?*
Questions?