Introduction to Corpora

Text Data and Machine Learning for Social Science

Elliott Ash

Text Data Course, Bocconi 2018

Overview

- These slides describe the process of getting a corpus of written language.
- Input:
  - A set of documents (e.g. text files), D.
- Output:
  - A matrix, X, containing statistics about phrase frequencies in those documents.

Text as Data

- Text data is a sequence of characters called **documents**.
- The set of documents is the **corpus**.
- Text data is **unstructured**:
  - the information we want is mixed together with (lots of) information we don't.
  - How to separate the two?
- All text data approaches will throw away some information:
  - The trick is figuring out how to retain valuable information.

Documents and metadata

- For small corpora, you might have the text and metadata together in a spreadsheet.
- For larger corpora, you might have:
  - A document is a text file (or an item in a relational database).
  - A corpus is a folder of text files.
  - The filenames for the text files should contain an identifier for linking to metadata.

What counts as a document?

- The unit of document analysis will vary depending on your question.
- If you are looking at how judges decide different types of cases, then a case would be a document.
- If you are looking at how judges differ within a court, then you might aggregate all of a judge's cases as a document.
- If you are looking at the impact of court cases on crime in a year, you might aggregate all the cases in a single year as a single document.
- If you are looking at how different topics are discussed within single cases, then a document might be a section or a paragraph.

Setting up Python and Jupyter

- Instructions for setting up Python, as well as links to all of the code examples, are linked from the syllabus.
  - Email me if you have problems.
- Course demonstrations will be done (and problem sets should be submitted) as Jupyter notebooks
  - see Geron, Chapter 2.
  - Navigate to your directory, and at terminal, type “jupyter notebook”
  - open a browser and click to `http://localhost:888/`
  - Click “New...” then “Python 3” to start a new notebook.

Pandas data-frames

```
# open dataset
import pandas as pd
df1 = pd.read_csv('death-penalty-cases.csv')
df1.head() # show top few lines of data
df1.info()
df1['court_id'].value_counts()

%matplotlib inline
df1.hist()
```

Iterating over documents in a data-frame

```
# make sure you are in the "code" directory
from utils import process_document

# iterate over rows and add to dictionary
processed = {}
for i, row in df1.iterrows():
    docid = row['cluster_id'] # doc identifier
    text = row['snippet']     # text snippet
    document = process_document(text) # process
    processed[docid] = document # add to dictionary
```

Iterating over documents in text files

```
# select all files in your directory
from glob import glob
fnames = glob('contracts/*txt') # selects files

# iterate over files
for fname in fnames:
    docid = fname[5:-4] # get docid from filename
    text = open(fname.read()) # read file as string
    document = process_document(text) # process
    processed[docid] = document # add to dictionary
```

Saving data in Python

- pandas makes it easy to save files:

```
pd.to_pickle(processed, 'processed_corpus.pkl')
```

- If you have a dataframe `df`, you can save it as a Python pickle, CSV, Excel spreadsheet, or Stata dataset:

```
df.to_pickle('dataset.pkl')
df.to_csv('dataset.csv')
df.to_excel('dataset.xlsx')
df.to_stata('dataset.dta')
```

- pandas can read all of these formats:
  - e.g. `pd.read_csv()`, `pd.read_pickle()`