# **MODULE 3** Working with Textual Data

## KEY TOOLS

**Python**

A programming language that is commonly used when working with data. Python has high-level data structures, is interpretive in nature, and has a relatively simply syntax.

## ACTIVITY: Teach your neighbor about text pre-processing

☞ ***Slide M3 - 11***

| Term | Definition |
|---|---|
| **Punctuation** | "The first choice a researcher must make when deciding how to preprocess a corpus is what classes of characters and markup to consider as valid text. The most inclusive approach is simply to choose to preprocess all text, including numbers, any markup (html) or tags, punctuation, special characters ($, %, &, etc), and extra white-space characters. These non-letter characters and markup may be important in some analyses (e.g. hashtags that occur in Twitter data), but are considered uninformative in many applications. It is therefore standard practice to remove them. The most common of these character classes to remove is punctuation." |
| **Numbers** | "While punctuation is often considered uninformative, there are certain domains where numbers may carry important information. For example, references to particular sections in the U.S. Code ('Section 423', etc.) in a corpus of Congressional bills may be substantively meaningful regarding the content legislation. However, there are other applications where the inclusion of numbers may be less informative." |
| **Lowercasing** | "Another preprocessing step taken in most applications is the lowercasing of all letters in all words. The rationale for doing so is that that whether or not the first letter of a word is uppercase (such as when that words starts a sentence) most often does not affect its meaning. For example, 'Elephant' and 'elephant' both refer to the same creature, so it would seem odd to count them as two separate word types for the sake of corpus analysis. However, there are some instances where a word with the same spelling may have two different meanings that are distinguished via capitalization, such as 'rose' (the flower), and 'Rose' the proper name." |

| | |
|---|---|
| **Stemming** | "The next choice a researcher is faced with in a standard text preprocessing pipeline is whether or not to stem words. Stemming refers to the process of reducing a word to its most basic form (Porter, 1980). For example the words 'party', 'partying', and 'parties' all share a common stem 'parti'. Stemming is often employed as a vocabulary reduction technique, as it combines different forms of a word together. However, stemming can sometimes combine together words with substantively different meanings ('college students partying', and 'political parties'), which might be misleading in practice." |
| **Stopword Removal** | "…some words, often referred to as "stop words", are unlikely to convey much information. These consist of function words such as 'the', 'it', 'and', and 'she', and may also include some domain-specific examples such as 'congress' in a corpus of U.S. legislative texts. There is no single gold-standard list of English stopwords, but most lists range between 100 and 1,000 terms." |
| **n-gram Inclusion** | "While it is most common to treat individual words as the unit of analysis, some words have a highly ambiguous meaning when taken out of context. For example the word 'national' has substantially different interpretations when used in the multi-word expressions: "national defense", and "national debt". This has led to a common practice of including n-grams from documents where an n-gram is a contiguous sequence of tokens of length n (Manning and Schutze, 1999). For example, the multi-word expression 'a common practice' from the previous sentence would be referred to as a 3-gram or tri-gram." |
| **Infrequently Used Terms** | "In addition to removing common stopwords, researchers often remove terms that appear very infrequently as part of corpus preprocessing. The rationale for this choice is often two-fold; (1) theoretically, if the researcher is interested in patterns of term usage across documents, very infrequently used terms will not contribute much information about document similarity. And (2) practically, this choice to discard infrequently used terms may greatly reduce the size of the vocabulary, which can dramatically speed up many corpus analysis tasks." |

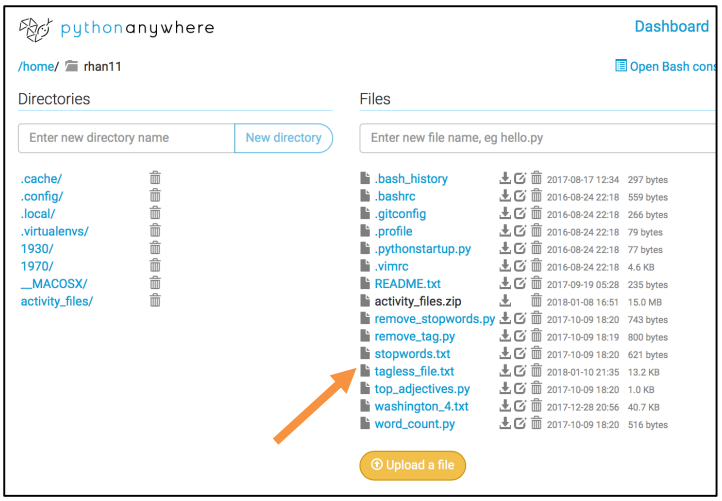
From: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2849145 (Denny and Spirling, 2017)

## ACTIVITY: Run a Python script to strip HTML tags from a text file

☞ ***Slide M3 - 17***

*In order to make our Washington speech file more useful, we'll need to remove the HTML tags.*

1. In your Bash shell type the command: `python remove_tag.py washington_4.txt`
2. Go back to the Dashboard by clicking on the "PythonAnywhere" logo in the top left corner.
3. Click on the "Browse files" button in the Files column Browse files Or click on the "Files" option in the upper right corner of the page.
4. In your list of files, you should now see a file named "tagless_file.txt". Click on it to preview your results. If it is too difficult to read the unwrapped text here, you can go to your Bash console and use "less tagless_file.txt" to view the file content in your console instead. Press "q" to quit viewing.
5. The "tagless_file.txt" file should contain a clean version of your web scraped text.




## ACTIVITY: Now you try!

☞ ***Slide M3 - 25***

- *Can you run the script [remove_stopwords.py] to remove the stopwords?*
  - Hint: the remove_tag.py script we just ran required only a single argument, which was for the file to remove the tags from (washington_4.txt). The remove_stopwords.py script requires three arguments in addition to the name of the script you want to execute:
    - The input file (your tagless file)
    - The list of stop words (stopwords.txt)
    - The output file name (you make this up!)
  - Remember to put a space between each argument.

  **python remove_stopwords.py tagless_file.txt stopwords.txt washington_4_stops_removed.txt**

- *Can you edit the stopwords.txt file to customize your list? Are there any stop words you think would be important to remove from this text?* **Yes. Edit stopwords.txt file.**

**Katie Rawson and Trevor Muñoz, "Against Cleaning"**

http://curatingmenus.org/articles/against-cleaning/

'When humanities scholars recoil at data-driven research, they are often responding to the reductiveness inherent in this form of scholarship. This reductiveness can feel intellectually impoverishing to scholars who have spent their careers working through particular kinds of historical and cultural complexity… From within this worldview, data cleaning is then maligned because it is understood as a step that inscribes a normative order by wiping away what is different. The term "cleaning" implies that a data set is "messy." "Messy" suggests an underlying order. It supposes things already have a rightful place, but they're not in it—like socks on the bedroom floor rather than in the wardrobe or the laundry hamper.'

**Discussion Questions:**

- *What does this piece suggest about the nuances of data cleaning?*
  **This may not always be the best choice. Cleaning the data can change the meanings. By removing punctuation, for example, makes meanings unclear (questions vs. statements). There are ethical issues involved with changing raw data. Depending on the nature of the original data, cleaning it could take away the original meaning or effectively silence some perspectives.**
- *What does "clean" imply?*
  **That working with the original data is bad.**
- *How might you talk to researchers on your campus who would be uncomfortable with the idea of clean vs. messy data?*