Digital Humanities in Practice

WEEK 7a-b

Processing Order of Text Cleaning Choices

Stage I:

- Select desired text segments from document(s).

Stage II:

- Apply basic latin filter, if enabled.
- Apply character drops, if any.

Stage III:

- Apply ReplaceWiths, if any. This is applied in the text sequence.

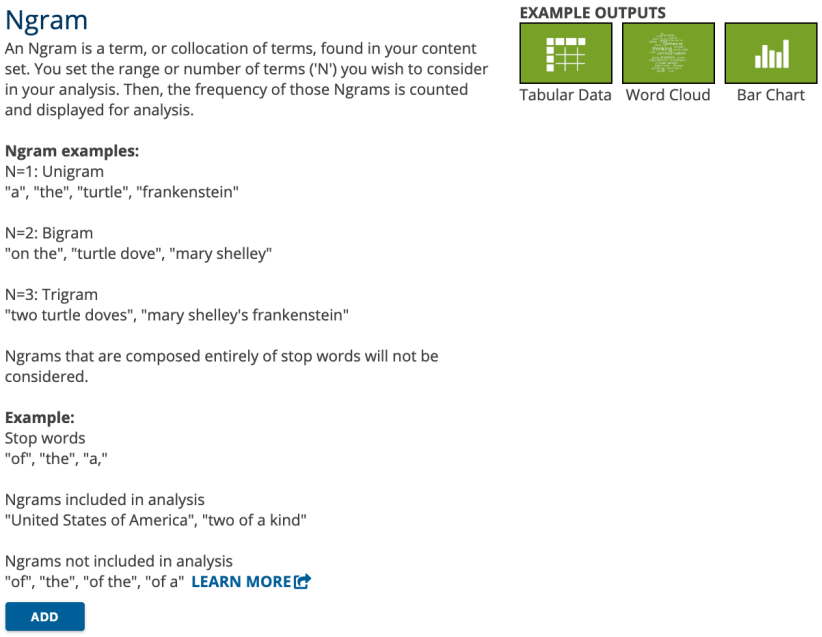
Stage IV:

- Apply stopword filter, if any.
- Apply multi-space squash if enabled.
- Apply lower-case, if enabled.

What comes out of Stage IV in the cleaning sequence is what is sent to any and all analysis jobs. None of the analysis tools have access to original document text. So, in the instance where we apply a clean configuration (with a stop word list containing "the" and "of") to an Ngrams job, is it correct to assume that "the" and "of" will be removed from "The United States of America"? Assuming "the" and "of" are included in the users stop words list and are filtered with correct (lower/upper) case, Then, yes, they will be removed. It may not be otherwise.

Ngrams

Ngrams build collocations of words from tokens within Documents of a Content Set. A Ngram is nothing more than a sequence of words, where N represents the number of words. A unigram has 1 word, bigram, 2, trigram, 3, and so on. Ngrams are often used to compile search terms or words that are often associated with one another. This tool can be used to help determine if a Content Set contains specific terminology, or phrase, which otherwise can be difficult to trace without intensive reading of each Document.



Digital Scholar Lab Implementation

The DSL leverages the Apache Lucene tokenizer to identify strings of words based on the configuration options you choose.

- The Ngrams Tool tokenizes on whitespaces following a basic analysis.
- Importantly, stop word lists are applied after the composition of Ngrams.
- Only Ngrams where all tokens appear in the stop word list is removed from the results: Ngrams containing one token, not in the stop word list are retained and output.
- The Tool permits users to prioritize occurrence by setting a minimum threshold for the number of times an Ngram must appear in order to be added to the resulting list.

- The Tool is case sensitive; e.g. ‘Apple’ and ‘apple’ are treated as distinct tokens for the purposes of Ngram analysis.
- There are two distinct types of visualizations associated with this Tool:
    - Word Cloud which employs Highcharts.com’s Word Cloud visualization.
    - Bar Chart which employs Highcharts.com’s Bar Chart visualization.

Readings

Jean-Baptiste Michel et al, " Quantitative Analysis of Culture Using Millions of Digitized Books." *Science* 331, 2011

Douglas Duhaime, 'Textual reuse in the Eighteenth Century: Mining Eliza Haywood's Quotations.” *DHQ*, 10:1, 2016 http://digitalhumanities.org:8081/dhq/vol/10/1/000229/000229.html

Maarten van den Bos, Hermione Giffard, “Mining Public Discourse for Emerging Dutch Nationalism.” *DHQ*, 10:3, 2016 http://digitalhumanities.org:8081/dhq/vol/10/3/000263/000263.html

Frederick W. Gibbs and Daniel J. Cohen. "A Conversation with Data: Prospecting Victorian Words and Ideas." *Victorian Studies*, vol. 54 no. 1, 2011, p. 69-77. *Project MUSE* muse.jhu.edu/article/468193. (I downloaded the article and it’s here)

Claude S. Fischer, “Digital Humanities, Big Data, and Ngrams.” *Boston Review* June 20, 2013. https://bostonreview.net/blog/digital-humanities-big-data-and-ngrams

M. Egnal, “Evolution of the Novel in the United States: The Statistical Evidence.” *Social Science History*, 37(2), 2013, p.231-254. doi:10.1017/S0145553200010646 (available here).

Example projects using ngrams

Google Ngram Viewer: https://books.google.com/ngrams

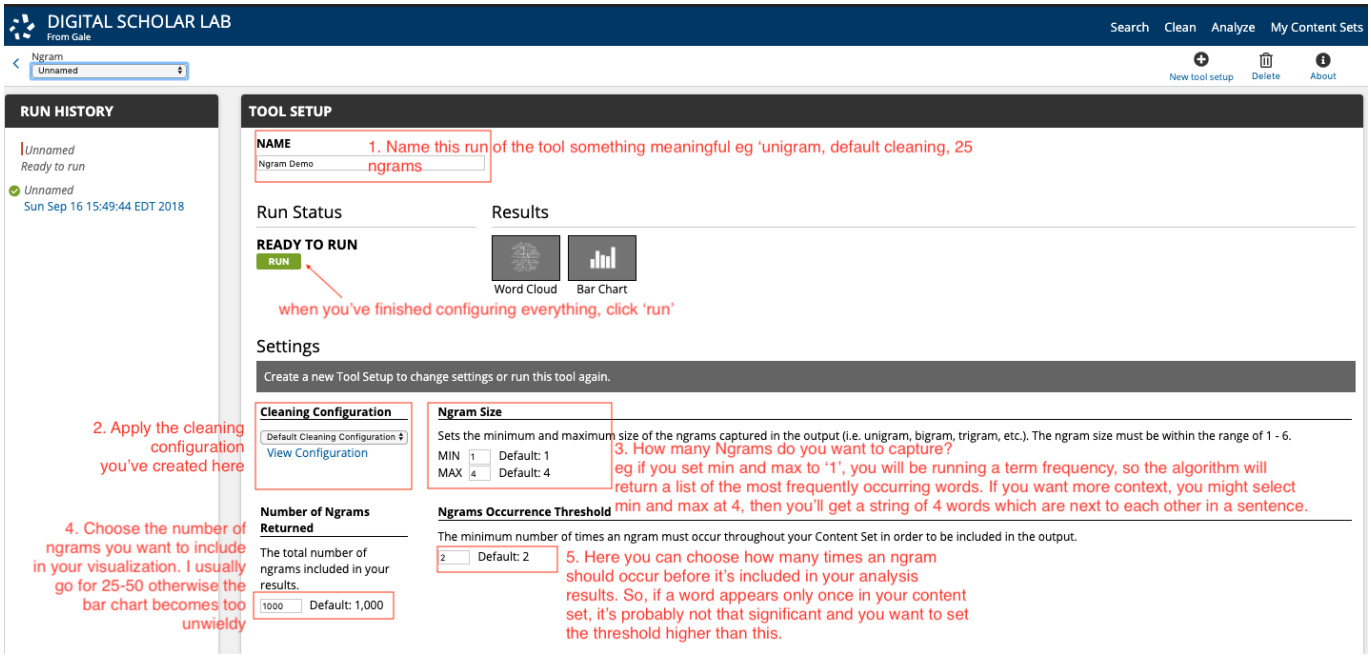
Voyant tools https://voyant-tools.org

Ben Schmidt, 'Poor man's sentiment analysis', *Sapping Attention*, Feb. 2 2012 https://sappingattention.blogspot.com/2012/02/poor-mans-sentiment-analysis.html

And projects described in the readings, above.

Configuration options in the DSL

Here are the options, also called out on the image, below.



1. Name this run of the tool something meaningful, eg 'unigram, default cleaning, 25 ngrams'

2. Apply cleaning configuration, if using.

3. How many ngrams do you want to capture? eg if you set min and max to '1', you will be running a term frequency, so the algorithm will return a list of the most frequently occurring words. If you want more context, you might select min and max at 4, then you'll get a string of 4 words which are next to each other in a sentence. The maximum n-gram size is 6.

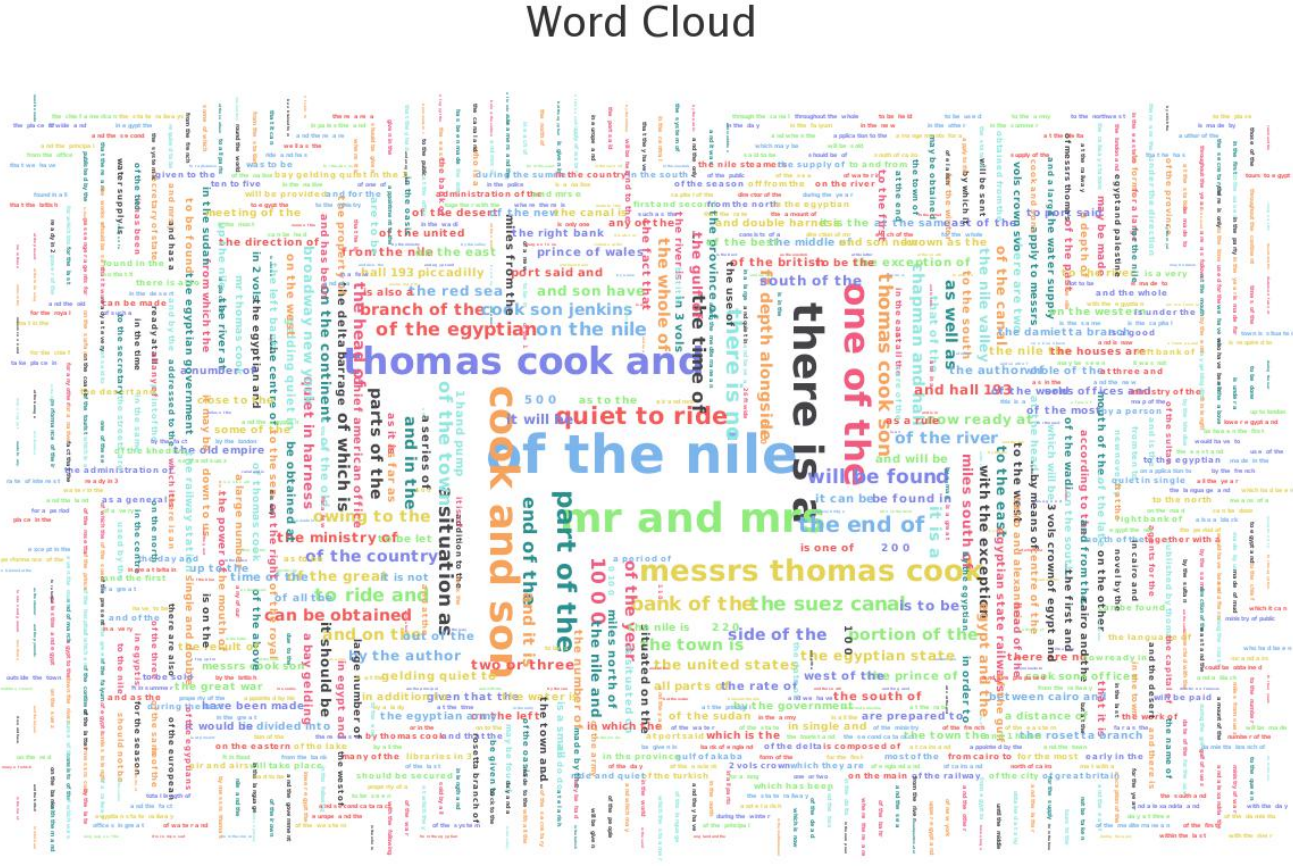
4. Choose the number of ngrams you want to include in your visualization. I usually go for 25-50 otherwise the bar chart becomes too unwieldy.

5. Here you can choose how many times an ngram should occur before it's included in your analysis results. So, if a word appears only once in your content set, it's probably not that significant and you want to set the threshold higher than this.

You have the option to view the results as either a bar chart or a word cloud. Both visualizations are downloadable.

- Word Cloud employs a visualization that presents Ngrams in a cluster, with those having the highest frequency or count in the center and those with lower counts towards the periphery of the visualization.
- Ngrams with higher counts are displayed with larger fonts, quickly drawing the viewer's attention to the most prominent feature as the most statistically relevant element of the visualization.
- The Bar Chart visualization represents the results as a standard Bar Chart, providing the viewer with counts of the occurrences of specific Ngrams in a Content Set.

Note: if you choose to look at collocates, a Word Cloud may not necessarily be the best visualization for these results! As you can see, it becomes almost unintelligible:



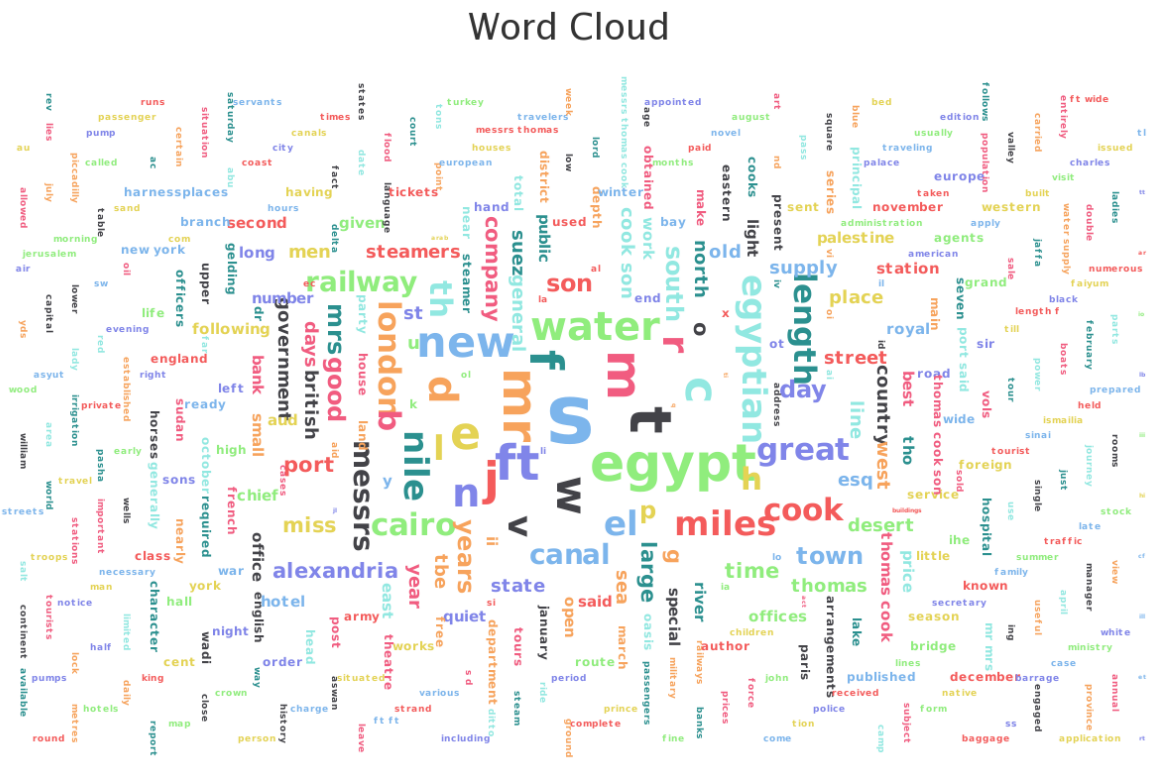
In this case, a downloaded CSV or bar graph will be more valuable. This underscores the importance of choosing the right sort of visualization for your results. For more commentary on Word Clouds, see

Jacob Harris, 'Word Clouds considered harmful', *Nieman Journalism Lab*, October 13 2011.

His use of the label 'mullets of the internet' to describe word cloud visualizations is more than enough reason to read the article in its entirety.

Using Ngrams as a tool for cleaning

You can include a simple term frequency analysis as part of your text cleaning workflow. Running the ngram tool with max and min thresholds set to '1', you will return a list of the most common words in your content set. Download the output as a CSV and open the file. You can cast your eye down the list, and copy any words that you don't want included in the final analysis. These may include 'nonsense words' created through poor OCR, word fragments, or words that you are simply not interested in analyzing for whatever reason.



ngram	count
s	742
t	516
egypt	463
mr	438
ft	423
m	420
e	411
f	407
j	403
c	402
new	375
w	373
water	350
d	349
n	339
r	326
l	293
nile	289
miles	288
egyptian	284
el	273
v	271
cairo	261
london	259
great	257
b	246
h	245
length	242
cook	241

Once you have copied your selected words, you can paste them into the stop words list on the ‘Clean’ page. Should you identify words which are frequently mis-recognized by the OCR engine, but you want to keep the correct spelling in your analyses, you can paste the misspelling into the ‘replace this’ box, and the correct spelling into the ‘with this’ box.