INTRODUCTION TO TEXT ANALYSIS WITH THE HATHITRUST RESEARCH CENTER

Exercises

Request an HTRC account by going to https://sharc.hathitrust.org and clicking on "Sign Up" in upper right-hand corner. Use an academic e-mail account.

I suggest using Google Chrome when working with the HTRC portal, as there's a tiny display bug with the Dunning Log Likelihood algorithm in Firefox. I've created two public worksets for you to reuse today, although feel free to reuse other public worksets, or to create your own. Limit your workset to 1-3 works for now to ensure you'll get results before the end of the workshop.

- *Frankenstein* (username: fgiannetti)
- *Dracula* (username: fgiannetti)

Task 1: Create a workset
Navigate to the HTRC Workset Builder (main portal > worksets > create workset). It requires a second login, but the credentials are the same as those you created for the main portal. After you log in, use the Blacklight interface to select your works, much as you would use a library catalog. Currently you can select from 3 million public domain volumes.

Task 2: Run either of these algorithms on *Frankenstein* and *Dracula* (or other small workset).

- Topic_Modeling
- Meandre_Tagcloud_with_Cleaning

Task 3: Run this comparative algorithm on the *Frankenstein* and *Dracula* worksets (or substitute other worksets that have some thread of commonality):

- Dunning_Log_likelihood_to_Tagcloud (Choose Frankenstein as the "analysis" workset and Dracula as the "reference" workset). After your job runs to completion, click on the result to see the job details of the completed job. Among the job details, you will see listed, among other file names, the two files dunning_tagcloud_over.html and dunning_tagcloud_under.html. Click on dunning_tagcloud_over.html. This will show you the words that are over-represented in the "analysis" workset (Frankenstein) compared to the "reference" workset (Dracula). Next click on dunning_tagcloud_under.html. This will show you the words that are under-represented in the "analysis" workset (Frankenstein) compared to the "reference" workset (Dracula). We will discuss and interpret the result!

Task 4: Explore Ngrams
Take some of the words you've identified from the previous steps and enter them into the HathiTrust Bookworm at http://bookworm.htrc.illinois.edu (you can add more than one word by pressing on the green + symbol). Pay attention to the frequency of those words through time, and be particularly mindful of their frequency when your chosen work was published.

Questions?
Francesca Giannetti, Digital Humanities Librarian, francesca.giannetti@rutgers.edu

Adapted from Green, Harriett and Sayan Bhattacharyya. "Introduction to the HathiTrust Research Center Portal (Version 3.0) for Text Mining Research." Workshop, Scholarly Commons, University of Illinois at Urbana-Champaign Library, 16 February 2015.

Names and brief descriptions of the algorithms available on the SHARC Portal (as of Jan 2016)

EF_Rsync_Script_Generator: Generate a script that allows you to download extracted features data for your workset of choice. The script can be run locally, listing the Rsync commands to access the volumes of the workset. For more information on the extracted features data see https://sharc.hathitrust.org/features.

MARC_downloader: Generates the set of MARC records for the elements of the workset. The result of running this algorithm is the creation of a zipfile (a compressed file). If you download the zipfile and then expand the zipfile (e.g. by clicking on its icon), you will see individual XML files, one for each volume that was in the workset, with each file consisting of a single MARC record (which is the MARC record of that volume).

Note: MARC is an abbreviation for Machine‑Readable Cataloguing Record, and refers to the bibliographical metadata that you typically see in a library catalog.

Classification_NaïveBayes: (Note: This algorithm does not currently work as expected.) The "Naïve Bayes" classifier is a simple classifier that classifies a particular object into an appropriate class (category) using a simple algorithm based on Bayes' Theorem from statistics. For example, suppose that a particular volume can belong to one of two classes, prose and poetry. The likelihood that a particular volume X (for which we don't know the classification) belongs to the class (category) "prose" could be computationally estimated using information present in the workset about the classes that other volumes in the workset belong to. This estimation can be carried out by an application of Bayes' Theorem, making use of the "prior" probabilities of any arbitrary volume in the workset being prose and poetry respectively, and "conditional" probabilities (the probabilities that an arbitrary volume in the workset is the particular volume X, on the condition, respectively, that its class is "prose" and "poetry".

Dunning_log‑likelihood_to_Tagcloud: Identifies words that are more common in one workset (the "analysis" workset) relative to another workset (the "reference" workset). Thus, it is useful for "contrasting" two worksets relative to each other. The resulting set of words is then visualized as a tag cloud.

OpenNLP_Date_Entities_to_Simile: Extracts date entities from a workset and displays them on a timeline. Each entity is displayed with the volume identifier (the unique identifier for that particular copy of the book within the HathiTrust Digital Library collection), the page identifier, and the sentence snippet within which it occurs.

OpenNLP_Entities_List: Extracts named entities (of the types specified by the user) from a workset, and presents the entities in a table. The user specifies a comma--‑separated list of entity types to be extracted, with acceptable types being: date, location, money, organization, percentage, person and time.

Spellcheck_Report_Per_Volume: Looks for and suggests replacements for errors in Optical Character Recognition (OCR). Generates a report containing the found instances.

Adapted from Green, Harriett and Sayan Bhattacharyya. "Introduction to the HathiTrust Research Center Portal (Version 3.0) for Text Mining Research." Workshop, Scholarly Commons, University of Illinois at Urbana-Champaign Library, 16 February 2015.

Meandre_Tagcloud: Displays the most frequent words in the workset as a "tag cloud", that is, with sizes of words proportionate to their frequencies.

Meandre_Tagcloud_with_Cleaning: Performs cleaning of the text before it allows you to create a tag cloud visualization of the most frequently occurring words in a workset. In a tag cloud, the size of the word is displayed in proportion to the number of times it occurred.

Topic_Modeling: Creates a list of "topics" from the workset. (A "topic", for the purposes of this algorithm, is simply a set of words that have a high probability of occurring in the vicinity of each other within the material in the workset.) Since a "topic" is simply a set of words, it is displayed as a "tag cloud".

Deployable_Word_Count: Displays the top N words in the workset, where N is specified by the user.

Adapted from Green, Harriett and Sayan Bhattacharyya. "Introduction to the HathiTrust Research Center Portal (Version 3.0) for Text Mining Research." Workshop, Scholarly Commons, University of Illinois at Urbana-Champaign Library, 16 February 2015.