How to Read Homer in 20 Minutes: The Script

This is the script for a twenty minute movie created for DH Unbound, and the movie describes and demonstrates some of the functionality of the Distant Reader Toolbox.

Download: Download the *Iliad* and the *Odyssey* from your favorite repository. I downloaded prose versions of the items from Project Gutenberg, and they were originally written/translated by Samuel Butler.

Print: Print the resulting files. Combined, the *Iliad* and the *Odyssey* are about 400 pages long. (For extra credit, bind the paper into a codex.) Use the traditional process to read the result. I'll wait.

Meanwhile: In the meantime, I'll remove the extraneous information (like the headers, footers, and introductions) from each each file, and divide the result into individual chapters. I'll then use the resulting files as input into any number of different software systems for analysis ("reading"). In this case, I will use a system called Distant Reader Toolbox.

Reader Toolbox: The Toolbox takes the input, does a whole lot of feature extraction against it, and saves the result as a set of delimited files and a relational database. The combination of these things is a data set, but I call them "study carrels". Put another way, the Toolbox takes a set of unstructured data (text) as input, applies both text mining and natural language processing against it, and outputs a set of structured data amenable to computation.

Build: Once a study carrel has been created, it can be modeled in quite a number of ways, and each model sheds a different light on the content. For example, we can learn about the carrel's extent -- its size and scope.

Extent: In this case, the carrel is comprised of 48 items totaling about 200,000 words, and has an average (Flesh) readability score of 76. Not very many items, about the same size as Melville's Moby Dick, and easier to read than not.

Bibliography: We can create a rudimentary bibliography from the carrel, and each includes a computed summary, a list of statistically significant words, and absolute pointers to a cache of original content. Like any bibliography, this can be used an a sort of index. Query the index for things of interest, peruse at the associated document.

Ngram analysis: Counting & tabulating the unigrams in the document can be informative, and visualizing the result tells some of a story. Since words are known by they company they keep, doing the same thing for bigrams tells a similar story.

collocations

Concordancing: The results of ngram analysis can be used as input to concordancing, and thus begin to answer the question, "How are these words used in context with other words?" Concordances are also useful for bringing to light well-known combinations of words. My favorites are nouns associated with forms of the word "be".

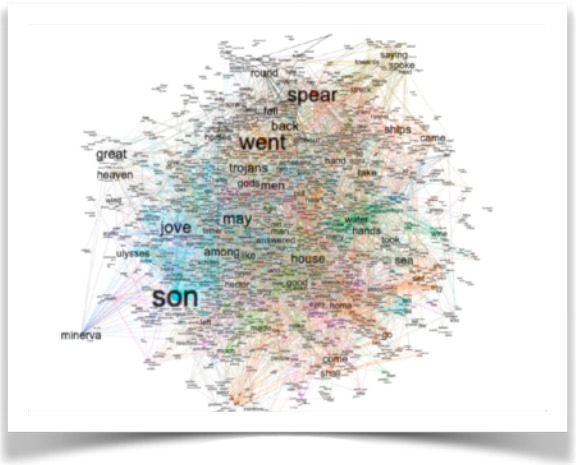
Computed keywords: Another way of identifying interesting words is to compute them. TF/IDF is a well-known technique, but

the Toolbox uses an algorithm called Yake. Again, the result can be illustrated with a word cloud.

Clustering: Clustering, as opposed to classification, is popular way to characterize textual corpora. Through the use of a dimension-reduction technique (Principle Component Analysis, or PCA), our carrel seems to fall into two distinct groups, but we don't know what they may be. Still, such an analysis points to what a good topic model might be.

Topic modeling: Topic Modeling, usually in the form of LDA, is another popular clustering technique. I usually start out small and work up. I also use the number of items in the corpus as a guide to the maximum number of topics.

Full text indexing: Reducing the features of a corpus down to a relational database, full indexing the result, and providing an expressive query language on top is yet another way to model narrative content. Using this technique all sorts of different views can be garnered.



collocations

Parts-of-speech: Language follows patterns. If it didn't, then we would not be able to understand each other. And so far, all of the modeling techniques can be applied to any language, sans any grammar. When modeling the text so each word is associated a parts-of-speech new questions can be asked and answered, such as, "What is is mentioned in this text, what do they do, and how are they described?"

Named entities: Nouns are a special case, and they can be modeled into named entities: people, places, organizations, dates, times, money amounts, etc. An analysis of these things addresses another set of questions, such as "What types of things exist in the corpus, and how are they related to each other?"

Grammars: Applying grammars to the text highlights additional patterns. The Toolbox supports a few grammars, and extracting sentences with a given grammar, it is possible to compare and contrast all sorts of things.

Collocations: Again, words are known by the company they keep, and exploiting collocations is one way to see what words are used in conjunction with other words.

Summary: The Distant Reader Toolbox consumes sets of unstructured data (text), applies text mining and natural language processing to the input, and saves the result as a data set. These data sets -- study carrels -- lend themselves to computing making it possible to answer all sorts of research questions.

`

Eric Lease Morgan <emorgan@nd.edu>
May 17, 2022