Where to start with text mining.

Posted on **August 14, 2012** by **tedunderwood**

[Edit June 8, 2015: This blog post has been rewritten and updated. See Seven Ways Humanists are Using Computers to Understand Text.]

This post is an outline of discussion topics I'm proposing for a workshop at NASSR2012 (a conference of Romanticists). I'm putting it on the blog since some of the links might be useful for a broader audience.

In the morning I'll give a few examples of concrete literary results produced by text mining. I'll start the afternoon workshop by opening two questions for discussion: first, what are the obstacles confronting a literary scholar who might want to experiment with quantitative methods? Second, how do those methods actually work, and what are their limits?

I'll also invite participants to play around with a collection of 818 works between 1780 and 1859, using an R program I've provided for the occasion. Links for these materials are at the end of this post.

I. HOW DIFFICULT IS IT TO GET STARTED?
There are two kinds of obstacles: getting the data you need, and getting the digital skills you need.

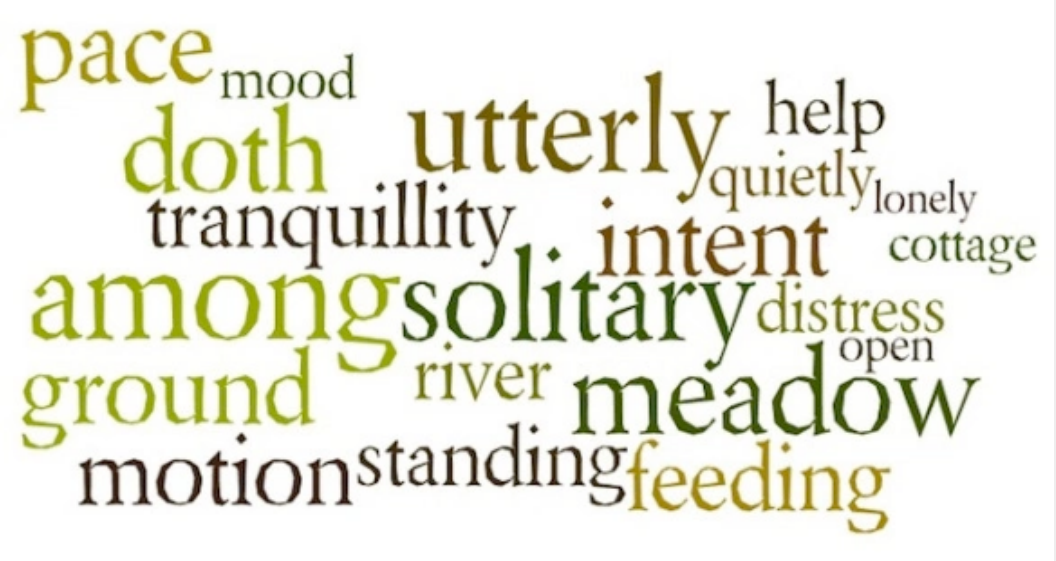
1. Is it really necessary to have a large collection of texts?
This is up for debate. But I tend to think the answer is "yes."

Not because bigger is better, or because "distant reading" is the new hotness. It's still true that a single passage, perceptively interpreted, may tell us more than a thousand volumes.

But if you want to interpret a single passage, you fortunately already have a wrinkled protein sponge that will do a better job than any computer. Quantitative analysis starts to

make things easier only when we start working on a scale where it's impossible for a human reader to hold everything in memory. Your mileage may vary, but I'd say, more than ten books?

And actually, you need a larger collection than that, because quantitative analysis tends to require context before it becomes meaningful. It doesn't mean much to say that the word "motion" is common in Wordsworth, for instance, until we know whether "motion" is *more* common in his works than in other nineteenth-century poets. So yes, text-mining can provide clues that lead to real insights about a single author or text. But it's likely that you'll need a collection of several hundred volumes, for comparison, before those clues become legible.



— Words that are consistently more common in works by William Wordsworth than in other poets from 1780 to 1850. I've used Wordle's graphics, but the words have been selected by a Mann-Whitney test, which measures overrepresentation relative to a context — not by Wordle's own (context-free) method. See the R script at the end of this post.

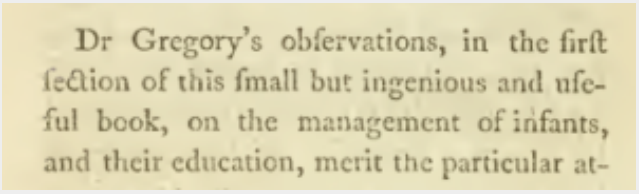
This isn't to deny that there are interesting things that can be done digitally with a single text: digital editing, building timelines and maps, and so on. I just doubt that *quantitative* analysis adds much value at that scale. (And to give credit where it's due: Mark Olsen was saying all this back in the 90s — see References.)

2. So, where do I get all those texts?

That's what I was asking myself 18 months ago. A lot of excitement about digital humanities is premised on the notion that we already have large collections of digitized sources waiting to be used. But it's not true, because page images are not the same thing as clean, machine-readable text.

If you're interested in twentieth-century secondary sources, the JSTOR Data for Research API can probably get you what you need. Primary sources are a harder problem. In our

own (Romantic) era, optical character recognition (OCR) is unreliable. The ratio of words transcribed accurately ranges from around 80% to around 98%, depending on print quality and typographical quirks like the notorious "long s." For a lot of text-mining purposes, 95% might be fine, if the errors were randomly distributed. But they're not random: errors cluster in certain words and periods.


Dr Gregory's obſervations, in the firſt
ſection of this ſmall but ingenious and uſe-
ful book, on the management of infants,
and their education, merit the particular at-

— What you see in a page image.

The problem can be addressed in several different ways. There are a few collections (like ECCO-TCP and the Brown Women Writers Project) that transcribe text manually. That's an ideal solution, but coverage of that kind is stronger in the eighteenth than the nineteenth century.

```
Dr Gregory's obfervations, in the firfi:
fe(5lion of this fmall but ingenious and ufe-
ful book, on the management of infants,
and their education, merit the particular at-
```

— What you may see as OCR.

So Jordan Sellers and I have supplemented those collections by automatically correcting 19c OCR that we got from the Internet Archive. Our strategy involved statistically cautious, period-specific spellchecking, combined with enough reasoning about context to realize that "mortal fin" is probably "mortal sin," even though "fin" is a correctly spelled word. It's not a perfect solution, but in our period it works well enough for text-mining purposes. We have corrected about 2,000 volumes this way, and are happy to share our texts and metadata, as well as the spellchecker itself (once I get it packaged well enough to distribute). I can give you either a zip file containing the 19c texts themselves, or a tab-separated file containing docIDs, words, and word counts for the whole collection. In either scheme, the docIDs are keyed to this metadata file.

Of course, selecting titles for a collection like this raises intractable questions about representativeness. We tried to maximize diversity while also selecting volumes that seemed to have reached a significant audience. But other scholars may have other priorities. I don't think it would be useful to seek a single right answer about representativeness; instead, I'd like to see multiple scholars building different kinds of collections, making them all public, and building on each other's work. Then we would be able to test a hypothesis against multiple collections, and see whether the obvious caveats about representativeness actually make a difference in any given instance.

3. Is it necessary to learn how to program?
I'm not going to try to answer that question, because it's complex and better addressed through discussion.

I will tell a brief story. I went into this gig thinking that I wouldn't have to do my own programming, since there were already public toolsets for text-mining (Voyant, MONK, MALLET, TAPoR, SEASR) and for visualization (Gephi). I figured I would just use those.

But I rapidly learned otherwise. Tools like MONK and Voyant taught me what was possible, but they weren't well adapted for managing a very large collection of texts, and didn't permit me to make my own methodological innovations. When you start trying to do either of those things, you rapidly need "nonstandard parts," which means that someone in the team has to be able to program.

That doesn't have to be a daunting prospect, because the programming involved is of a relatively forgiving sort. It's not easy, but it's also not professional software development. So if you want to do it yourself, that's a plausible aspiration. Alternately, if you want to collaborate with someone, you don't necessarily need to find "a computer scientist." A graduate student or fellow humanist who can program will do just fine.

If you do want to learn to program, I would recommend starting with either Python or R. Of the two languages, Python is certainly easier. It's intuitive, and well-documented, and great for working with text. If you expect to use existing tools (like MALLET), and just need some "glue" to connect them to each other, Python is probably the way to go. R is a more specialized and less intuitive language. But it happens to be specialized in some ways that are useful for text mining. In particular, it has built-in statistical functions, and a built-in plotting/graphing capacity. I've used it for the sample exercise that accompanies this post. But if you're learning to program for the first time, Python might be a better all-around choice, and you could in principle extend it to do everything R does. [Later addition: You could do worse than start with The Programming Historian.]

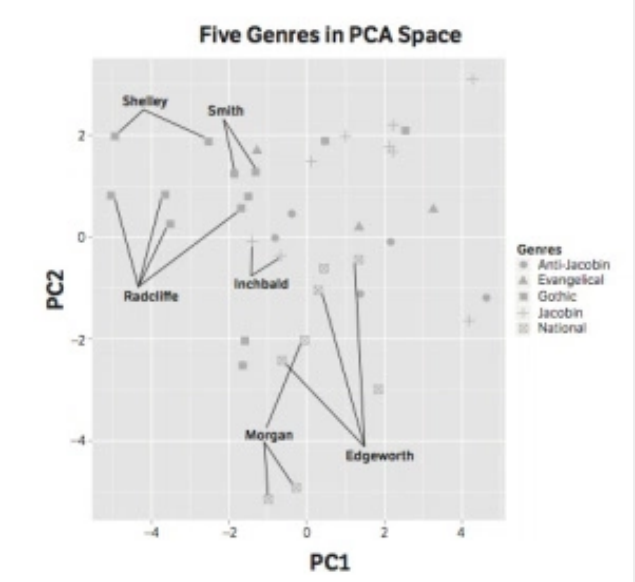
II. WHAT CAN WE ACTUALLY DO WITH QUANTITATIVE METHODS?
What follows is just a list of elements. Interesting research projects tend to combine several of these elementary operations in ad-hoc ways suited to a particular question. The list of elements runs a little long, so let me cut to the chase: the overall theme I'm trying to convey is that you can build complex arguments on a very simple foundation. Yes, at bottom, text mining is often about counting words. But a) words matter and b) they hang together in interesting ways, like individual dabs of paint that together start to form a picture.

So, to return to the original question: what can we do?

1) Categorize documents. You can "categorize" in several different senses.

> a) Information retrieval: retrieve documents that match a query. This is what you do every time you use a search engine.

b) (Supervised) classification: a program can learn to correctly distinguish texts by a given author, or learn (with a bit more difficulty) to distinguish poetry from prose, tragedies from history plays, or "gothic novels" from "sensation novels." (See "Quantitative Formalism," Pamphlet 1 from the Stanford Literary Lab.) The researcher has to provide examples of different categories, but doesn't have to specify *how* to make the distinction: algorithms can learn to recognize a combination of features that is the "fingerprint" of a given category.



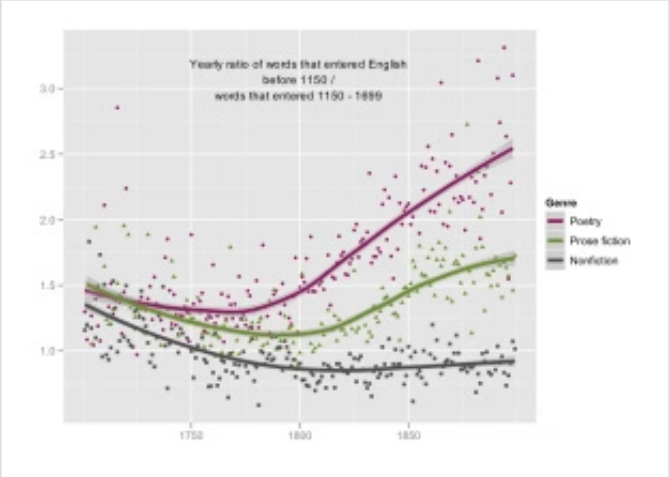
— An example of clustering from "Quantitative Formalism," Allison, Heuser, Jockers, Moretti, and Witmore, Stanford Literary Lab.

c) (Unsupervised) clustering: a program can subdivide a group of documents using general measures of similarity instead of predetermined categories. This may reveal patterns you don't expect.

All three of these techniques can achieve amazing results armed with what seems like very crude information about the documents they're categorizing. We know, intuitively, that merely counting words is not enough to distinguish a tragedy from a history play. But our intuitions are simply wrong — see the lit lab pamphlet I cited above. It turns out that there's an enormous amount of information contained in relative word frequencies, even if you know nothing about sequence or syntax. As you consider other aspects of text mining, it's useful to keep this intuitive misfire in mind. Relatively simple statistical techniques often characterize discourse a good deal better than our intuitions would predict.

2) Contrast the vocabulary of different corpora. In a way, this reverses the logic of classifying documents (1b). Instead of using features to sort documents into categories, you start with two categories of documents and contrast them to identify distinctive features.

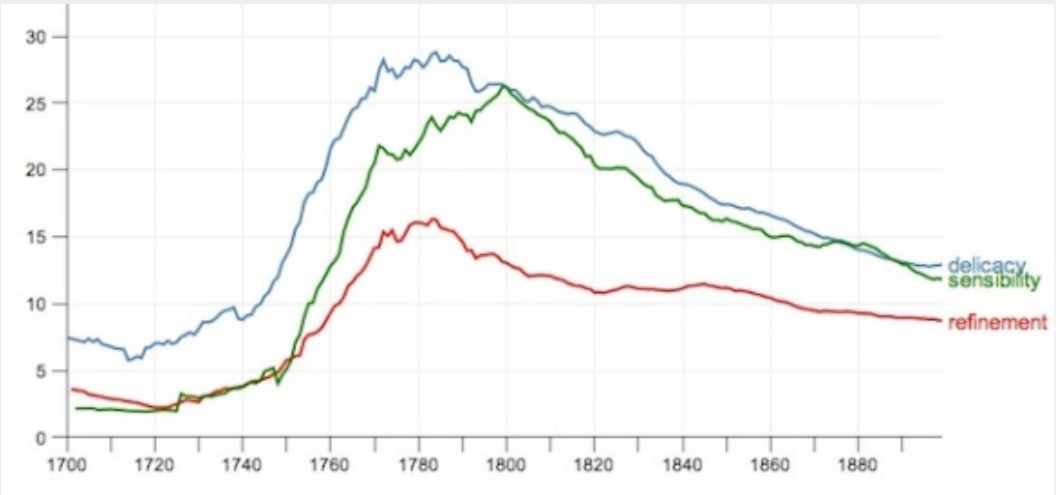
For instance, you can discover which words (or phrases) are overrepresented in one author or genre (relative to, say, the rest of nineteenth-century literature). It can admittedly be a challenge to interpret the results: this is a kind of evidence we aren't accustomed to yet. But lists of overrepresented words can be a fruitful source of critical leads to pursue in more traditional ways.



Beyond identifying distinctive words and phrases, corpora can be compared using metrics chosen for some more specific reason. It's difficult to give an exhaustive list – but, for instance, the argument I've been making about generic differentiation is based on a kind of corpus comparison. As a general think-piece on the topic, I recommend Ben Schmidt's blog post arguing that comparison is an underused and underrated tool; Schmidt's taxonomy of text-mining techniques in that post was a strong influence on the taxonomy I'm offering here.

3) Trace the history of particular features (words or phrases) over time. This could be viewed as a special category of corpus comparison, where you're comparing corpora segmented on the time axis.

The best-known example here would be Google's ngram viewer. Digital humanists love to criticize the ngram viewer, partly for valid reasons (there's no way to know what texts are being used). But it has probably been the single most influential application of text mining, so clearly people are finding this simple kind of diachronic visualization useful. A couple of other projects have built on the same dataset, slicing it in different ways. Mark Davies of BYU built an interface that lets you survey the history of collocations. Our team at Illinois built an interface that mines 18-19c correlations in the ngram dataset; it turns out that correlated words have a high likelihood of being related in other ways as well, and these can be intriguing leads: see what words correlate with "delicacy" in our period, for instance. Harvard has built Bookworm, which can be understood as a smaller but more flexible and better-documented version of the ngram viewer (built on the Open Library instead of Google Books).



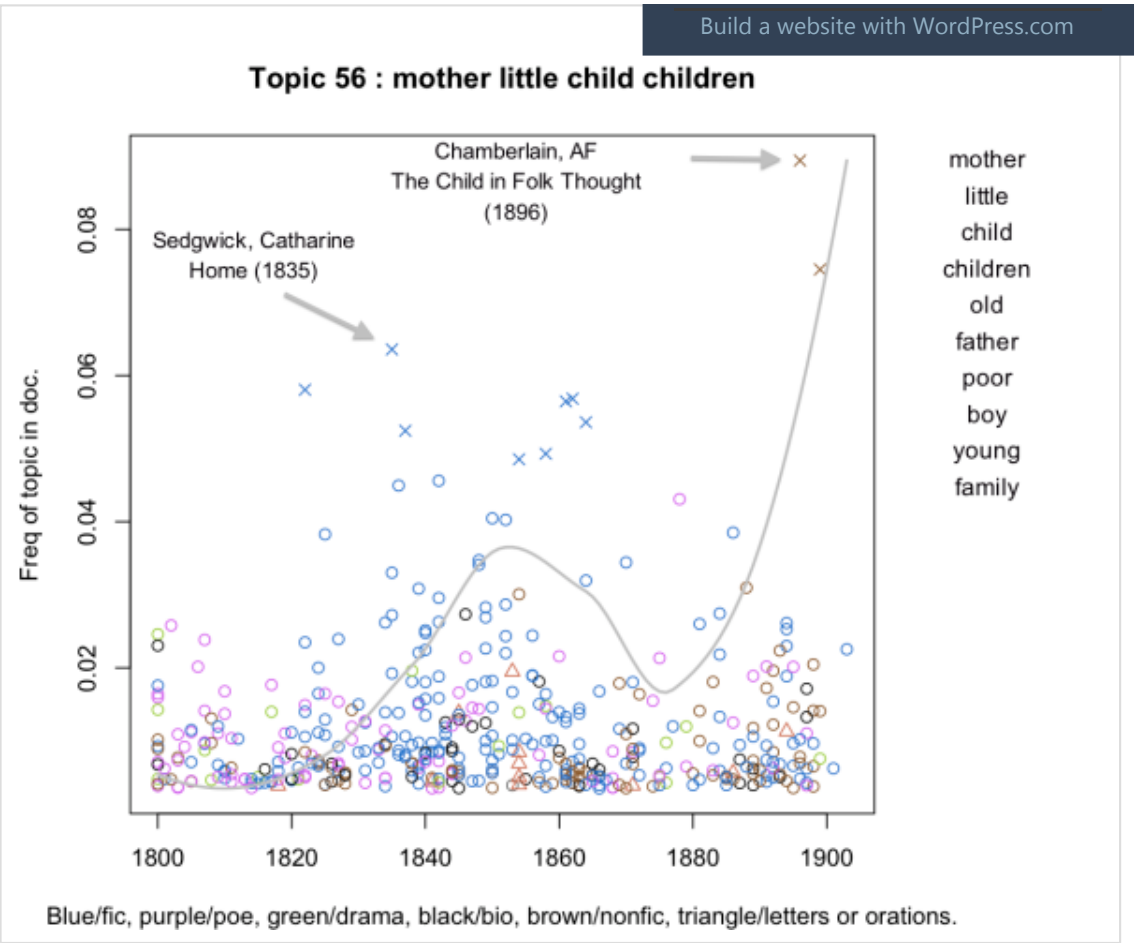
— Words whose frequencies correlate strongly over time are often related in other ways as well. Ngram viewer by Auvil, Capitanu, Heuser and Underwood, based on corrected Google dataset.

Of special interest to Romanticists: a project that isn't built on the ngram dataset but that does use diachronic correlation-mining as a central methodology. In Stanford Lit Lab Pamphlet 4, Ryan Heuser and Long Le-Khac have traced some very interesting, strongly correlated changes in novelistic diction over the course of the 19th century.

Finally, anyone who wants to make a diachronic argument about diction should read Ben Schmidt's simple, elegant experiment peeling apart two different components of change: generational succession and historical change within the diction of a single age-cohort.

4) Cluster features that tend to be associated in a given corpus of documents (aka topic modeling). In a way, this reverses the logic of clustering documents (1c). Instead of grouping documents that tend to share the same words, you group words that tend to appear in the same documents, or parts of documents. This produces something that looks like a semantic map of the period or corpus you're studying. (It would be more accurate to call it a discursive map, because topics don't actually have to be unified semantically. They are more analogous to "discourses.")

There are a lot of ways to cluster features, ranging f
Semantic Analysis), to the new, hip approach — "Ba
advantage that it clusters individual occurrences of
As a result, it can distinguish different senses of a w
clear and comprehensive introduction to topic model



Topic modeling has become justifiably popular for several reasons. First and foremost, a "discursive map" can be a nice thing to have; it lends itself easily to interpretation. Also, frankly, this approach doesn't require a whole lot of improvisation. You just pour text files into a tool like MALLET, and out come a list of topics, looking meaningful and authoritative. It's important to remember that topic-modeling is in fact an imprecise process. Slightly different inputs (for instance, a different stopword list) can produce very different outputs.

5) Entity extraction. If you're mainly interested in proper nouns (personal names or place names, or dates and prices) there are tools like OpenNLP that can extract these from text, using syntactic patterns as clues.

6) Visualization. Perhaps this isn't technically a form of analysis, but in practice it's important enough that it deserves to be treated as a separate analytical step. It's impractical to list all possible forms of visualization here, but for instance, results can be visualized:

> a) Geographically — to reflect, for instance, density of references to different parts of the world. (See a quick example of Matt Wilkens' excellent work in this mode — and a more complete argument available through this conference program.)
>
> b) As a network graph — to reflect strength of affinity between different entities (characters, or topics, or what have you).

c) Through "Principal Component Analysis," if you have multidimensional data that need to be flattened to two dimensions for ease of comprehension.

Putting things together.
There's no limit to the number of ways you can combine these different operations. Matt Wilkens has extracted references to named entities from fiction, and then visualized their density geographically. Robert K. Nelson has performed topic modeling on the print run of a Civil-War-era newspaper, and then graphed the frequency of each topic over time. You could go a step further and look for correlations between topics (either over time, or in terms of their distribution over documents). Then you could visualize the relationships between topics as a network.

What's the goal uniting all this experimentation? I suspect there are two different but equally valid goals. In some cases, we're going to find patterns that actually function as evidence to support literary-historical arguments. (In a number of the examples cited above, I think that's starting to happen.) In other cases, text mining may work mainly as an exploratory technique, revealing clues that need to be fleshed out and written up using more traditional critical methods. The boundary between those two applications will be hotly debated for years, so I won't attempt to define it here.

III. SAMPLE DATA AND SCRIPT FOR EXPLORATION.
I don't know whether we'll really have time for this, but I ought to at least offer you a chance to do hands-on stuff. So here's a medium-sized project.

I've created a pre-packaged set of 818 volumes of poetry and fiction between 1780 and 1859, including 243 authors. I can give you first, a metadata file that includes the authors, titles, dates, and so on for each volume, and second, a data file that includes word counts for each volume. (To keep from frying your laptop, I've only included the top 9,000 words in the collection. But actually that's a lot.)

Finally, I've provided an R script that will let you define different chunks of the collection and compare them against each other, to identify words that are significantly overrepresented in a given author, genre, or period. The script will try two different measures of "overrepresentation": the first, "log-likelihood," is based on the aggregate frequency of words in the corpus you selected, adding all the volumes in the corpus together. The second, "Mann-Whitney rho," tries to locate words that are *consistently* more common in corpus X by paying attention to individual volumes. For more on how that works, see this blog post.

Of course, the R script won't work until you download R and open it from within R. Please understand that this is a very rough, ad-hoc piece of work for this one occasion, not a polished piece of software that I expect people to use for the long term.

Postscript about the word "mining."
I know it has an industrial sound; I know humanists like "analysis" more. But I'm sticking with the mining metaphor on the principle of truth in advertising. I think that word

accurately conveys the scale of this enterprise, and the fact that it's often more exploratory than probative. Besides, "mining" is vivid, and that has its own sort of humanistic value.

References (that aren't already implicit in links)

Mark Olsen, "Signs, Symbols, and Discourses: A New Direction for Computer-Aided Literature Studies" Computers and the Humanities 27 (1993): 309-314.

RELATED

For most literary scholars, text mining is going to be an exploratory tool.
In "18c"

Seven ways humanists are using computers to understand text.
In "disciplinary history"

We don't already understand the broad outlines of literary history.
In "fiction"

This entry was posted in **18c**, **19c**, **collection-building**, **Romantic-era writing**, **visualization** by **tedunderwood**. Bookmark the **permalink [https://tedunderwood.com/2012/08/14/where-to-start-with-text-mining/]** .

About tedunderwood

Ted Underwood is Professor of English at the University of Illinois, Urbana-Champaign. On Twitter he is @Ted_Underwood.
View all posts by tedunderwood →

50 THOUGHTS ON "WHERE TO START WITH TEXT MINING."

Pingback: Where to start with text mining « Another Word For It

disgruntledphd
on **August 16, 2012 at 1:02 pm** said:

Ted,

Thank you for this article. I'm a psychologist (with a PhD in progress) and am familiar with R. I looked at your code, and well done, it looks like a really nice way to introduce someone to R and text analytics. If you haven't already, you might want to see Unix for poets (see http://people.sslmit.unibo.it/~baroni/compling04/UnixforPoets.pdf) as its an eminently readable introduction to the use of small command line tools for text analysis (or mining). I have found that R somewhat suffers in a text mining sense, as the massively sparse matrices coupled with the in memory approach causes problems when one wants to examine a lot of text. In any case, thanks for the post, and best of luck on your analytic adventures.

tedunderwood
on **August 16, 2012 at 2:38 pm** said:

I agree. Didn't have time to get into it here, but I believe you're right that R has some scalability issues. I hear about people using R on Hadoop, but I don't know how that works. In my own work, I actually tend to couple R with MySQL — and increasingly, Java — so that in effect I'm using R to visualize the end product of analysis that is mostly done in other languages. But when I say that, it sounds complicated, and in this post I'm trying not to scare people off … so shhhh ….

G Lau
on **March 12, 2013 at 4:45 am** said:

Hi Ted,

Your post came up as I was searching for some solutions on text mining/analytics in R. Specifically I was trying to find out if someone out there already has a clever method to dealing with the sparsity issue when text mining in R.

You mentioned using R with MySQL and Java, any chance of a new post on a framework for analyzing large corpus in with R/MySQL?

Cheers!

tedunderwood
on **March 12, 2013 at 7:29 am** said:

> Thanks! My workflow with R/MySQL is modeled on Ben Schmidt's workflow, and I believe he may have a post somewhere explaining how he does it. The basic story is that storing texts as "sparse tables" in MySQL addresses the sparsity problem (at least where storage space is concerned).
>
> Usually when I draw data out of MySQL to work in R, I limit either the # of words or the # of documents. If I have to work with a large number of documents, I'll usually select only the top 1000 or 5000 words.
>
> I'm not sure if these pragmatic details are the "sparsity issues" that bother you, or whether you're confronting the bigger problem they call the "curse of dimensionality." I don't have any novel solution to that one!

Pingback: Links – August 18, 2012 | zota

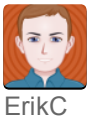
Pingback: Week 1: mapping businesses auxiliary to the Georgian satirical print trade | cradledincaricature

Waqar Jaffry
on **September 8, 2012 at 4:09 pm** said:

Dear Ted,

Thanks for sharing your thoughts. It is a wonderful expression.

Pingback: On Simplicity » the scottbot irregular | e-Map Store | Online Map | Digital Map



ErikC
on **September 23, 2012 at 9:09 am** said:

Reblogged this on Erik Champion and commented:
Intro to txt mining!

Pingback: shinenkan

Pingback: Data Mining/Distant Reading/Visualisation (Reading Reflection, Week 6) « travellingyuezhi

Pingback: Proposed Grad Course for 2013-14: Digital Romanticisms (Draft) » Roger T. Whitson, Ph.D

Dale
on **October 7, 2012 at 1:18 am** said:

Anyone you can suggest who could hep with a text mining project?
Any suggestion would be welcome.

Pingback: Jane Austen and contemporary prose style | Pages and Lights

RD
on **January 28, 2013 at 10:05 am** said:

Hi! I'm analizing text with R. The first thing I am interested is simple frequences of all one-word concepts in text. I am using tm package. R seems to not stem many simple words like "feel", "feeling", which I would prefer to see as "feel". Or even quite simple plural forms like "game" and "games". Is this the best I can get from embeded stemming without additional programming by myself? What do you use for such task?

tedunderwood
on **January 28, 2013 at 11:56 am** said:

To be honest, I use very little stemming myself. I've played around with tools that do stem or lemmatize words (The Monk Project can do this, for instance). But in my experience, it hasn't been very helpful, and I can think

of many cases where the information discarded by stemming turns out to be important. (Verb tenses, for instance, tell you a lot about genre.) If you absolutely do need stemming, you might use the NLTK module in Python (http://nltk.org/api/nltk.stem.html). I'm not aware of an R module that does stemming, though one might exist.

> 
> **Romans Dinuls**
> on **January 28, 2013 at 12:58 pm** said:
>
> Thnx! I was searching throught other forums also and it looks like stemming is not so advanced yet. There is obviously many things to be done to imporve user's control on what information is consolidated and how it's done.
> I will check NTLK anyway.


rhymerchick
on **January 30, 2013 at 12:19 am** said:

Reblogged this on DigIn' the Humanities! and commented:
Here is a blog entry on "Where to start with text mining" that looks very useful!

Pingback: RegEx and other gibberish | Course Notes--Digital Humanities Methods and Pract

Pingback: Is this what reading looks like?


Ryan
on **June 27, 2013 at 1:12 am** said:

I need help with a simple text mining application for a new business venture in the online learneducation anding space and would appreciate some feedback from individuals on their interest in being involved or some broader solutions to my problem.

Pingback: Voyant Tools, Teaching Edition, DH2013, Lincoln, Nebraska (July 2013) – Voyant Tools Documentation

Pingback: What Digital Humanists Do |

Pingback: Beginners in the Digital Humanities | THATCamp St. Louis 2013

Latuji Jnr.
on **November 30, 2013 at 8:10 am** said:

Reblogged this on My Research Collections.

Jennifer H
on **January 8, 2014 at 2:26 pm** said:

Thank you ; this is incredibly helpful!

Pingback: Evans-TCP: What it is and How Early Americanists Might Use It |

Jim Greenberg (@greenbjb)
on **February 27, 2014 at 12:54 pm** said:

Ted,

Thanks for the thoughtful post. We are at the early stages of integrating text analysis into our undergraduate social science (Political Science, Sociology) programs. It turns out to be a pretty heavy lift. These tools take time to install, configure, learn, and peel at to make them accessible to undergrads. Do you know of anyone that has developed an undergraduate course that introduces students to text analysis? Perhaps even one that is designed with social science students in mind? Thanks again.

tedunderwood
on **February 27, 2014 at 1:06 pm** said:

Thanks for the note. I agree that this is a pretty heavy lift for an undergrad methods class. I don't right off the bat know of *undergraduate* syllabi that have integrated this kind of thing effectively yet. Matt Wilkens has this graduate syllabus, but that's pitched at a higher level and also not trying to cover everything you might need in an undergrad social-science methods class.

I can imagine doing it, though. Topic modeling with Mallet isn't too hard; that could be taught at an undergrad level. Analyzing the data afterward is harder for humanists, because we don't generally have stats, but in my fantasies about social-science methods classes they already include a tool like R or Matlab for statistical data analysis. Is that just a fantasy?

dave
on **April 12, 2014 at 1:59 pm** said:

What did you use to visualize the words, where the size of the word reflects some metric? (i.e. in your William Wordsworth figure)

Pingback: Scotland's Collections and the Digital Humanities

Pingback: Not Just Books | Reading Review for September 23

Pingback: Text Analysis in Digital Humanities | Connie's Digital History

Pingback: Text Mining and Visualizing History | History Notes 2.0

Pingback: Big Data and a Metallica Database: From A Qualitative Approach to a Failed Research Method - &

Pingback: Text Analysis | Digital History

Pingback: Draft for New Course: Digital Toolbox for Historians (UNR) | Christopher M. Church

Pingback: Links: DH Posts I Find Useful | Plans Gone Awry

Pingback: Digital Textual Analysis | Rob's Blog

Pingback: 5/21 Agenda; 5/28 Assignments | Medill Digital Frameworks Class

Pingback: 5/28 Agenda; 6/4 Assignments | Medill Digital Frameworks Class

Pingback: 6/4 Agenda; 6/9 (Last Day!) Assignments | Medill Digital Frameworks Class

Pingback: Agenda 8/13; Assignments 8/20 | Medill Digital Frameworks Class

Pingback: Agenda 8/20; Assignments 8/25 | Medill Digital Frameworks Class

Pingback: For Text Mining beginners | 2IT422 Data Engineering

Pingback: Create a More User Friendly Webpage | Enterprise Search

Pingback: Full-Service Consulting for Business Excellence | CRM Consulting

Pingback: Intro. to DH Syllabus Draft – Kirstyn Leuner | Digital Romanticisms

Pingback: Digital Textual Analysis – The Digital Past: My Blog

Comments are closed.