ARTICLES

D-Lib Magazine
March 2006

Volume 12 Number 3

ISSN 1082-9873

From Babel to Knowledge

Data Mining Large Digital Collections

Daniel J. Cohen
George Mason University
<dcohen@gmu.edu>

In Jorge Luis Borges's curious short story *The Library of Babel*, the narrator describes an endless collection of books stored from floor to ceiling in a labyrinth of countless hexagonal rooms. The pages of the library's books seem to contain random sequences of letters and spaces; occasionally a few intelligible words emerge in the sea of paper and ink. Nevertheless, readers diligently, and exasperatingly, scan the shelves for coherent passages. The narrator himself has wandered numerous rooms in search of enlightenment, but with resignation he simply awaits his death and burial – which Borges explains (with signature dark humor) consists of being tossed unceremoniously over the library's banister.

Borges's nightmare, of course, is a cursed vision of the research methods of disciplines such as literature, history, and philosophy, where the careful reading of books, one after the other, is supposed to lead inexorably to knowledge and understanding. Computer scientists would approach Borges's library far differently. Employing the information theory that forms the basis for search engines and other computerized techniques for assessing in one fell swoop large masses of documents, they would quickly realize the collection's incoherence though sampling and statistical methods – and wisely start looking for the library's exit. These computational methods, which allow us to find patterns, determine relationships, categorize documents, and extract information from massive corpuses, will form the basis for new tools for research in the humanities and other disciplines in the coming decade.

For the past three years I have been experimenting with how to provide such end-user tools – that is, tools that harness the power of vast electronic collections while hiding much of their complicated technical plumbing. In particular, I have made extensive use of the application programming interfaces (APIs) the leading search engines provide for programmers to query their databases directly (from server to server without using their web interfaces). In addition, I have explored how one might extract information from large digital collections, from the well-curated lexicographic database WordNet to the democratic (and poorly curated) online reference work Wikipedia. While processing these digital corpuses is currently an imperfect science, even now useful tools can be created by combining various collections and methods for searching and analyzing them. And more importantly, these nascent services suggest a future in which information can be gleaned from, and sense can be made out of, even imperfect digital libraries of enormous scale. A brief examination of two approaches to data mining large digital collections hints at this future, while also providing some lessons about how to get there.

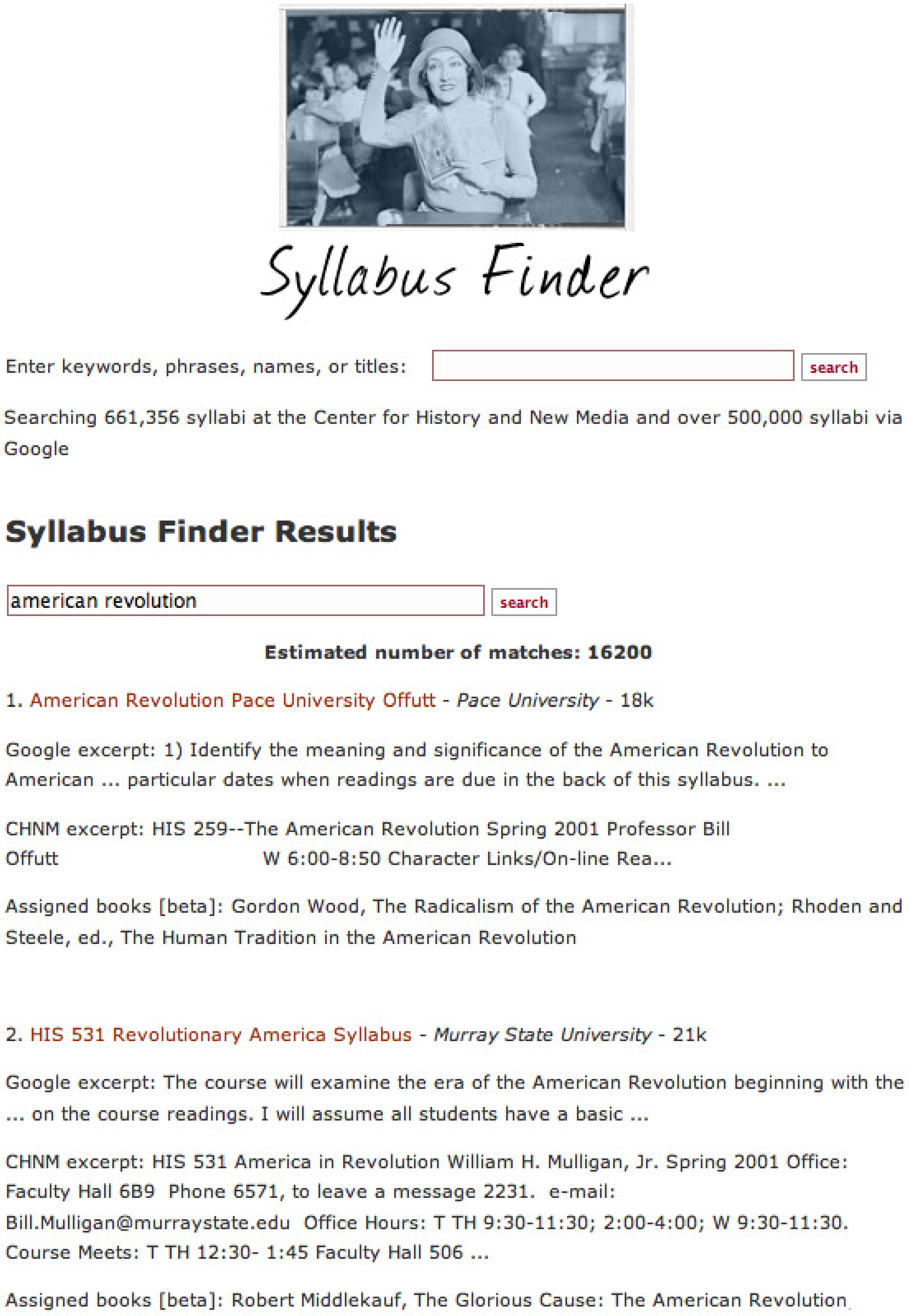
Document Classification: Syllabus Finder

A deceptively simple example of what one can already do with access to vast numbers of documents and some programming is my Syllabus Finder.[1] When Google released its web search API in the spring of 2002,[2] I decided to experiment with how one might build a specialized search engine to find course materials on the web, since these documents were increasingly appearing online but had not been (and are not likely to be) centralized in a single repository. I thought a search engine that could locate syllabi on any topic would be useful for professors planning their next course, for discovering the kinds of books and assignments being commonly assigned, and for understanding the state of instruction more broadly.[3]

Computer scientists call this problem of finding similar documents like syllabi in a large corpus "document classification." While there are many techniques for pursuing such categorization, I chose one of the oldest and most basic, called "keyword-in-context indexing" (KWIC), pioneered by Hans P. Luhn, an erstwhile textile mill manager and polymath who before joining IBM in 1941 pioneered methods for organizing cocktail recipes (among other inventions). Cocktail recipes normally had each drink's name at the top, followed by the list of ingredients; Luhn's system created a separate set of cards that listed each *ingredient* at the top, followed by each cocktail containing that ingredient – what would later be known as an inverted index, the basis for many information retrieval techniques. At IBM in the 1950s, Luhn set his sights slightly higher than simplified bartending and discovered that documents on the same topic had what he called "dictionaries of notions," or a core of identifying keywords that are much more likely to show up in these documents than in the general universe of documents. In the last half-century, computer scientists focusing on informational retrieval have greatly improved upon Luhn's (and his colleagues') algorithms, but many of the fundamental processes remain the same.[4]

Applying Luhn's methodology to generate a "dictionary of notions" for syllabi was fairly straightforward. I first downloaded the text from a relatively small set of history syllabi and parsed them for individual words, which I then ranked by frequency. Looking at this frequency list, I discovered that beyond the obvious, common words that show up on most web pages (and English-language documents in general, such as "the," "and," etc.) there were certain words that appeared very frequently on syllabi, far more so than on a typical web page. This list included, of course, "syllabus" (appearing on over 90% of syllabi), "readings" (and its singular, "reading"), "assignment" (and its plural "assignments"), "exam" and its variants and synonyms, and the word "week." When I ran the same analyses for word couplets, I found the most common pairings were "fall" and "spring" followed by a four-digit number (the "Spring 2006" that is found on so many syllabi this term) followed by the all-important "office hours."

The Syllabus Finder tool I subsequently began to build simply reversed this logic: A web page that contains many of these words – basic ingredients in the pedagogical cocktail – is extremely likely to be a syllabus. Users of this specialized search engine enter desired topics in a basic form, and the software then sends specially optimized queries to Google's API service that include words from the dictionary of "syllabus" terms to maximize the possibility that the results set will include actual syllabi (it also simultaneously queries a database of syllabi stored locally). Since Google's API returns its search results in a SOAP envelope (one of the main XML schemas used for server-to-server communications), the Syllabus Finder software can easily convert these possible matches into programming objects or arrays and then pass them on to statistical analyses and secondary intra-document searches (such as extractors that use regular expressions and other text analysis methods to pull out the college or university the syllabus is from as well as its assigned books). In short, if you simply ran some queries manually on the Google home page, you would not find as many syllabi as through the Syllabus Finder, nor would you see the additional information the specialized search engine provides.



Enter keywords, phrases, names, or titles:

Searching 661,356 syllabi at the Center for History and New Media and over 500,000 syllabi via Google

Syllabus Finder Results

american revolution

Estimated number of matches: 16200

1. American Revolution Pace University Offutt - *Pace University* - 18k

Google excerpt: 1) Identify the meaning and significance of the American Revolution to American ... particular dates when readings are due in the back of this syllabus. ...

CHNM excerpt: HIS 259--The American Revolution Spring 2001 Professor Bill Offutt W 6:00-8:50 Character Links/On-line Rea...

Assigned books [beta]: Gordon Wood, The Radicalism of the American Revolution; Rhoden and Steele, ed., The Human Tradition in the American Revolution

2. HIS 531 Revolutionary America Syllabus - *Murray State University* - 21k

Google excerpt: The course will examine the era of the American Revolution beginning with the ... on the course readings. I will assume all students have a basic ...

CHNM excerpt: HIS 531 America in Revolution William H. Mulligan, Jr. Spring 2001 Office: Faculty Hall 6B9 Phone 6571, to leave a message 2231. e-mail: Bill.Mulligan@murraystate.edu Office Hours: T TH 9:30-11:30; 2:00-4:00; W 9:30-11:30. Course Meets: T TH 12:30- 1:45 Faculty Hall 506 ...

Assigned books [beta]: Robert Middlekauf, The Glorious Cause: The American Revolution

Like the mathematics of KWIC, the Syllabus Finder isn't perfect; rapidly handling highly variable, heterogeneous, poorly formatted and tagged documents dispersed across the web, it occasionally returns documents that are not syllabi. But it does a surprisingly good job at achieving its modest goal – on most topics for every ten documents it retrieves, about nine are syllabi – and it has thus far found and catalogued over 600,000 syllabi, synthesizing a collection of course materials considerably larger than any created or maintained by a professional organization, educational institution, or library, or by any other effort on the web to aggregate syllabi.[5]

Beyond suggesting the power of using search APIs to retrieve relevant documents and subject them to further automated analysis or combine them with sources retrieved from other locations simultaneously, the Syllabus Finder also suggests an important role in the future for open-access reference materials and corpora. These resources can be leveraged to better scan, sort, and mine other digital collections that are unwieldy because of their scale or lack of metadata. For example, say you have a billion unstructured, untagged, unsorted documents related to the American presidency in the last twenty years. How would you differentiate between documents about George H. W. Bush (Sr.) and George W. Bush (Jr.)? This is a tough document classification problem because both presidents are often referred to as just "George Bush" or "Bush," or, even worse, "the President." However, by using Yahoo's Term Extraction Web service (part of its content analysis toolset that extracts significant words and phrases from documents for KWIC purposes), you could pull out of encyclopedia entries for the two Bushes the most common words and phrases that were likely to show up in documents about each (e.g., "Berlin Wall" and "Barbara" vs. "September 11" and "Laura"). You would still run into some disambiguation problems ("Saddam Hussein," "Iraq," and "Cheney" would show up a lot for both), but this method is actually quite a powerful start to document classification.[6]

I tested this theory by writing a small program to combine Yahoo's Term Extraction API with a popular online reference source. I then ran the program on the main entries for the two Bush presidencies, and received these terms:

George H. W. Bush	George W. Bush
president bush	president bush
saddam hussein	office of homeland security
fall of the berlin wall	reconciliation act
tiananmen square	internal revenue service
thanksgiving day	irs
american troops	department of veterans affairs
manuel noriega	congress
halabja	franklin d roosevelt
invasion of panama	ronald reagan
gulf war	terrorist attacks
help	war on terror
saudi arabia	aftermath
united nations	military spy
berlin wall	military airport
	chinese military
	public aspect
	spy plane
	democratic parties
	security office
	approval rating

To be sure, some odd terms appear on these two lists (one of the few photos on the entry for the first President Bush highlights him visiting the troops in Saudi Arabia on Thanksgiving Day in 1990; following 9/11, the second President Bush had the highest approval rating for any president since Franklin Delano Roosevelt), and the maximum document size on Yahoo's service meant that the full entries were truncated, leading to an overemphasis on the early part of each presidency (obviously a higher maximum would have resulted in less emphasis on the 2001 brouhaha with the Chinese over an errant spy plane, now almost completely forgotten). Yet even this imperfect list would probably suffice for a good first cut at dividing a large collection of documents.

Furthermore, it may come as a surprise that the encyclopedia entries scanned to create such lists do not have to be perfect – only fairly reliable and openly available on the Internet. Indeed, the reference source I used for this experiment was Wikipedia, the democratically written encyclopedia much disparaged by publishers and professors. Despite its flaws, however, Wikipedia will probably do just as well for basic KWIC profiling of document classes as the *Encyclopaedia Britannica*.

Of course, it would be even better to have a Web service without the document size limitation imposed by Yahoo operating on a more rigorous corpus than Wikipedia, one that included the more obscure regions of academic research. Such a valuable service could rapidly speed up existing types of manual searches that scholars do. For instance, KWIC document classification, or better yet the more sophisticated statistical text categorizers behind email spam filters (such as those that use probabilistic word models based on Bayes' theorem), could swiftly locate most of the love letters in a large digital archive or all of the lab notes on stem cells in a biological database, even if the words "love" or "stem cells" did not appear on some of those documents.

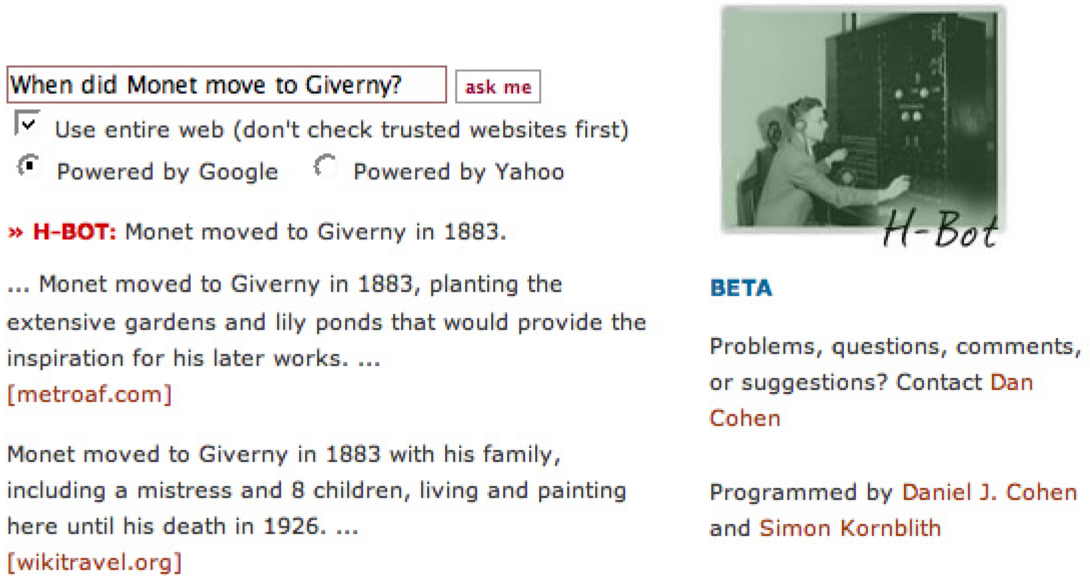
Question Answering: H-Bot

Although far from the highest form of the discipline, much of history deals with answering factual questions about the past. In which year was Charles Darwin's *Origin of Species* published? (1859.) When did French peasants storm the Bastille? (July 14, 1789.) Who was the first president of Liberia? (Joseph Jenkins Roberts, an émigré from Virginia.) In the digital age, answering such questions has become vastly easier, and indeed this process even has the potential to be automated, freeing up historians and their students to engage in more advanced analyses of the past.[7] Here again, computer scientists have a great deal to teach the humanists, who are used to consulting reference sources manually or scanning primary and secondary sources for answers. Indeed, "question answering" (QA), like document classification, has been a staple of information theory courses and information retrieval software for decades. Because the automated extraction of answers from large corpuses has promising commercial and defense (i.e., intelligence) applications, for the last fourteen years the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense have sponsored the Text Retrieval Conference (TREC) to spur development in this area.[8]

QA is a far greater challenge than document classification because it exercises almost all of the computational muscles. Not only do you have to find relevant documents in massive corpuses (involving methodologies of search and document classification), you also have to interpret users' questions well to know what they are looking for (natural language processing) and analyze the text of retrieved documents using a variety of statistical and linguistic methods (information theory, regular expressions, and other text parsing techniques).

The experimental tool I have worked on in the field of QA is called H-Bot, an automated historical fact finder (programmed with Simon Kornblith).[9] H-Bot accepts natural language queries about historical events, people, and terms, and then uses a series of data-mining techniques to try to answer them. For instance, it can rapidly answer simply phrased questions such as "When did Charles Lindbergh fly to Paris?" Although it has a fast mode that looks at "trusted sources" first (i.e., online encyclopedias and dictionaries), it can also use the entire Web to answer questions through text analysis and algorithms. To determine when Charles Lindbergh took his famous flight to Paris (in this "pure" mode, where it does not simply try to find a reliable encyclopedia entry), H-Bot uses an open source natural language processing package and associated dictionaries to pull out the key terms "Charles

Lindbergh" and "Paris." It also uses a freely available English irregular verb lookup table to convert "fly" into the more useful (for historical purposes) "flew." H-Bot then asks Google to send it a cache of web pages that have all of those words, which the software then (in the case of questions that begin with "when") scans for strings that look like years (i.e., positive three- and four-digit numbers). In this case it finds many instances of "1902" and "1974" (Lindbergh's birth and death years), but, most of all, it finds a statistically indicative spike around "1927," the year that Lindbergh made his pioneering flight to Paris.



When did Monet move to Giverny?

☑ Use entire web (don't check trusted websites first)

● Powered by Google ○ Powered by Yahoo

» **H-BOT:** Monet moved to Giverny in 1883.

... Monet moved to Giverny in 1883, planting the extensive gardens and lily ponds that would provide the inspiration for his later works. ...
[metroaf.com]

Monet moved to Giverny in 1883 with his family, including a mistress and 8 children, living and painting here until his death in 1926. ...
[wikitravel.org]

BETA

Problems, questions, comments, or suggestions? Contact Dan Cohen

Programmed by Daniel J. Cohen and Simon Kornblith

For complex questions, H-Bot takes advantage of more advanced algorithms and mathematical equations from information theory. For instance, a version of H-Bot (currently unavailable to the public) attempts to answer multiple-choice history questions by extracting significant words in the question and each of the possible answers, and then running statistical analyses via the Google API to see how frequently each set of terms appear together on the Web. This determination of the "normalized information distance" between terms, or how closely related they seem to be in the universe of meaning, allows H-Bot to successfully answer, for instance, a U.S. National Assessment of Educational Progress exam question asking who started the Montgomery bus boycott. The software understands through its frequency calculations that "Rosa Parks," rather than "Phyllis Wheatley," "Mary McLeod Bethune," or "Shirley Chisholm," is the correct answer.[10]

Lessons

The Syllabus Finder and H-Bot, like the Web from which they extract information, may be imperfect, but they show what can be done by stitching together and processing digital collections using server-to-server communications and programming algorithms. My early experience in building these digital tools has also afforded me three initial, related lessons – lessons that are not entirely in accord with key premises of some of those working in the world of digital libraries.

1. *More emphasis needs to be placed on creating APIs for digital collections*. Since the 1960s, computer scientists have used application programming interfaces to provide colleagues with robust, direct access to their databases and digital tools. Access via APIs is generally far more powerful than simple web-based access. APIs often include complex methods drawn from programming languages – precise ways of choosing materials to extract, methods to generate statistics, ways of searching, culling, and pulling together disparate data – that enable outside users to develop their own tools or information resources based on the work of others. In short, APIs hold great promise as a method for combining and manipulating various digital resources and tools in a free-form and potent way.

Unfortunately, even after four decades APIs remain much more common in the commercial realm – for example, the APIs provided by Google and Yahoo – than in the nonprofit sector. There are some obvious reasons for this disparity. By supplying an API, the owners of a resource or tool generally bear most of the cost (on their taxed servers, in technical support and staff time) while receiving little or no (immediate) benefit. Moreover, by essentially making an end-run around the normal or "official" ways of accessing a collection, such as a web search form for a digital archive, an API may devalue the hard work and thoughtfulness put into the more public front end.

So why should nonprofit digital collections provide APIs, especially given their often limited funding compared to a Google or Yahoo? The reason IBM conceived APIs in the first place, and still today the reason many computer scientists find APIs highly beneficial, is that unlike other forms of access they encourage the kind of energetic and creative third-party development that in the long run – after the initial costs borne by the API's owner – maximize the value and utility of a digital resource or tool. Motivated by a variety of goals and employing disparate methodologies, users of APIs often take digital resources or tools in directions completely unforeseen by their owners. APIs have provided fertile ground for thousands of developers to experiment with the tremendous indices, services, and document caches maintained by Google and Yahoo, as hundreds of sites creating map "mashups" have done recently. New resources based on APIs appear weekly, some of them hinting at new methods for digital research, data visualization techniques, and novel ways to data-mine texts and synthesize knowledge.[11]

APIs need not be complex. Owners of digital collections can create a rudimentary API by repackaging their collection's existing search tool using simple Web services protocols such as REST (Representational State Transfer), where a URL sent to a server returns an XML document rather than an HTML results page. Users of the API can then parse the XML document to extract the information they need or would like to combine with (or pass on to) other services.

2. *Resources that are free to use in any way, even if they are imperfect, are more valuable than those that are gated or use-restricted, even if those resources are qualitatively better.* The techniques discussed in this article require the combination of dispersed collections and programming tools, which can only happen if each of these services or sources is openly available on the Internet. Why use Wikipedia, which can be edited – or vandalized – by anyone? Not only can one send out a software agent to scan the entire Bush articles on the Wikipedia site (whereas the same spider is turned away by the gated *Encyclopaedia Britannica*), one can instruct a program to *download* the entire Wikipedia and store it on one's server (as we have done at the Center for History and New Media), and then subject that corpus to more advanced manipulations. While flawed, Wikipedia is thus extremely valuable for data-mining purposes. For the same reason, the Open Content Alliance digitization project (involving Yahoo, Microsoft, and the Internet Archive, among others) will likely prove more useful for advanced digital research than Google's far more ambitious library scanning project, which only promises a limited kind of search and retrieval.[12]

3. *Quantity may make up for a lack of quality*. We humanists care about quality; we greatly respect the scholarly editions of texts that grace the well-tended shelves of university research libraries and disdain the simple, threadbare paperback editions that populate the shelves of airport bookstores. The former provides a host of helpful apparatuses, such as a way to check on sources and an index, while the latter merely gives us plain, unembellished text. But the Web has shown what can happen when you aggregate a very large set of merely decent (or even worse) documents. As the size of a collection grows, you can begin to extract information and knowledge from it in ways that are impossible with small collections, even if the quality of individual documents in that giant corpus is relatively poor. If you want to know when Charles Lindbergh crossed the Atlantic you can look at many Web pages and realize that most of them say 1927 (even if a few say 1926 or 1928). Or you can write a program to automate this process using some crafty algorithms, ranking

schemes, and text analysis. The larger the collection such a program operates on, the fewer errors it will make.

In other words, high-quality digitization and thorough text markup may be attractive for those creating digital collections, but a familiarity with information theory and data-mining techniques makes one realize that it may be more worthwhile to digitize a greater number of books or documents at a lower standard for the same cost. In Borges's imagined (and nightmarish) library, no two books are the same, and very few repeat even the smallest of phrases. In our libraries – once analog and now digital – we come across countless similar phrases, wordings, and facts in numerous books, and many books refer to each other through footnotes and bibliographies. This repetition and cross-referencing should allow us to create tools for mining the vast information and knowledge that lies within the nearly limitless digital collections we are about to encounter.

Acknowledgements

The author would like to thank Roy Rosenzweig for his helpful comments on an earlier draft of this article.

Notes

1. Hosted at the Center for History and New Media at <http://chnm.gmu.edu/tools/syllabi/>.

2. Google, *Google Web APIs*, <http://www.google.com/apis/>.

3. For an example of this analysis, see Daniel J. Cohen, "By the Book: Assessing the Place of Textbooks in U.S. Survey Courses," *Journal of American History* 91 (March 2005), 1405-1415.

4. For the origins of the method I use, see H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development*, 1(4):309-317, October 1957; Harold Borko and Myrna Bernick, "Automatic Document Classification." *Journal of the ACM*, 10(2):151-162, April 1963.

5. It is worth comparing the Syllabus Finder with the OpenCourseWare Finder at <http://opencontent.org/ocwfinder/>. The latter always returns syllabi, but because it requires syllabi to conform to the OCW standard and format, it locates far fewer total syllabi.

6. Yahoo, *Content Analysis Web Services: Term Extraction*, <http://developer.yahoo.net/search/content/V1/termExtraction.html>.

7. Daniel J. Cohen and Roy Rosenzweig, "No Computer Left Behind," *Chronicle of Higher Education*, 24 February 2006, B6-8.

8. National Institute of Standards and Technology, *Text REtrieval Conference*, <http://trec.nist.gov>. For the latest on this conference, see NIST, "Overview of the TREC 2004 Question Answering Track," <http://trec.nist.gov/pubs/trec13/papers/QA.OVERVIEW.pdf>. For a similar methodology to that used by H-Bot, see Hristo Tanev, Milen Kouylekov, and Bernardo Magnini, "Combining Linguistic Processing and Web Mining for Question Answering: ITC-irst at TREC," <http://trec.nist.gov/pubs/trec13/papers/itc-irst-tanev.web.qa.pdf>. For some earlier foundations of this methodology, see Bernardo Magnini et al., "Is it the Right Answer? Exploiting Web Redundancy for Answer Validation," Association for Computational Linguistics 40th Anniversary Meeting (ACL-02), of Pennsylvania, Philadelphia, July 7 - 12; Bernardo Magnini et al., "Comparing Statistical and Content-Based Techniques for Answer Validation on the Web," *Proceedings of the VIII Convegno AI*IA*, Siena - Italy, September

11-13. Bernardo Magnini et al., "Mining Knowledge from Repeated Co- occurrences," *TREC-11 Conference Notebook Papers*, Gaithersburg, MD, November 19-22.

9. Available in beta at the Center for History and New Media at <http://chnm.gmu.edu/tools/h-bot/>.

10. For the origins of this theory, see A.N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems in Information Transmission*, volume 1, number 1 (1965), pp. 1-7. For its modern application, see Paul Vitanyi and Rudi Cilibrasi, "Automatic Meaning Discovery Using Google," at http://arxiv.org/abs/cs/0412098, accessed on 30 April 2005. For more on this topic, see Daniel J. Cohen and Roy Rosenzweig, "Web of Lies? Historical Knowledge on the Internet," *First Monday* 10 (December 2005), <http://firstmonday.org/issues/issue10_12/cohen>.

11. At the Center for History and New Media we have recently launched a digital collection related to the hurricanes of 2005, which uses the Google Maps API to locate each photo, story, or multimedia file on the Gulf Coast, providing a new way of searching historical archives. Center for History and New Media, *Hurricane Digital Memory Bank*, <http://hurricanearchive.org>.

12. Open Content Alliance, <http://www.opencontentalliance.org/>; Google Book Search, <http://books.google.com>.

Top | Contents
Search | Author Index | Title Index | Back Issues
Previous Article | Next article
Home | E-mail the Editor

doi:10.1045/march2006-cohen