Exploring a Digital Library through Key Ideas

Bill N. Schilit and Okan Kolak
Google Research
1600 Amphitheatre Parkway,
Mountain View, CA 95054
{schilit,okan}@google.com

ABSTRACT

Key Ideas is a technique for exploring digital libraries by navigating passages that repeat across multiple books. From these *popular passages* emerge quotations that authors have copied from book to book because they capture an idea particularly well: Jefferson on liberty; Stanton on women's rights; and Gibson on cyberpunk. We augment Popular Passages by extracting key terms from the surrounding context and computing sets of related key terms. We then create an interaction model where readers fluidly explore the library by viewing popular quotations on a particular key term, and follow links to quotations on related key terms. In this paper we describe our vision and motivation for Key Ideas, present an implementation running over a massive, real-world digital library consisting of over a million scanned books, and describe some of the technical and design challenges. The principal contribution of this paper is the interaction model and prototype system for browsing digital libraries of books using key terms extracted from the aggregate context of popularly quoted passages.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; J.5 [**Arts and Humanities**]: Literature; H.4.3 [**Information Systems Applications**]: Communications Applications—*Information browsers*

General Terms

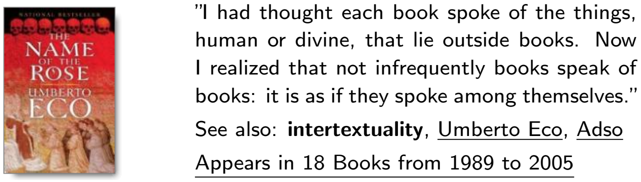
Algorithms, Design, Human Factors

Keywords

digital libraries; quotations; humanities research; data mining; key phrases; hypertext; great ideas.

1. INTRODUCTION

In our research we pose the question: How can we help people engage online books in a style similar to browsing online encyclopedia? Information technology's progress towards "an increasing democratization or dissemination of power" (Landow [10]) requires a ready supply of authoritative voices. Expanding access to these knowledge sources is extremely important in a world where technology broadcasts one voice to many, authors are anonymous, credibility is ambiguous, views are potentially biased, and where "we have people who are trying to repeatedly abuse our sites" (Wales [13]).



"I had thought each book spoke of the things, human or divine, that lie outside books. Now I realized that not infrequently books speak of books: it is as if they spoke among themselves."

See also: **intertextuality**, Umberto Eco, Adso

Appears in 18 Books from 1989 to 2005

Figure 1: One of 700-plus popularly quoted passages appearing under the key term "intertextuality." Tens of millions of quotations emerge from a clean-slate data mining algorithm that takes a digital library of scanned books as input. The reader follows "See also" links to explore related quotations or "Appears in *n* Books..." links to read the quote in context in primary and secondary source books (example modified slightly for readability).

Our larger aim is to expose the general population to the great ideas and the connections between ideas that lie within books. Mining and linking ideas are both significant challenges. Mining requires that we distinguish words about "ideas" from other words that appear in books, and, at the surface, all words look alike. Recognizing relations between ideas is difficult because there is no common classification system. Author's use the *same* words to name *different* ideas, and *different* words to name the *same* idea. Acknowledging the large problem scope and extensive efforts that preceded ours, the research presented here should be taken as a report on a new path we have been exploring rather than a complete solution.

Umberto Eco writes that "books speak of books: it is as if they spoke among themselves" (see Figure 1). With apologies to Postermodernists, this sums up our approach. We developed a language-independent data mining capability to extract quotations[1] from books. Data mining provides

JCDL'08, June 16–20, 2008, Pittsburgh, Pennsylvania, USA.
Copyright 2008 ACM 978-1-59593-998-2/08/06 ...$5.00.

[1] A quotation is a passage, of a certain length, repeated across books. A passage that is short is more likely a col-

the text of each quotation, all the books where the quotation appears, as well as the context (surrounding text) of each use. We use quotations to create links between book passages thereby giving readers the ability to jump between pages where quotations occur.

Linking book passages is only one benefit of mining quotations. In general when one author employs another's words we can say the following: (1) the passage reflects an idea or describes an event particularly well (or, by and large, better than passages not-quoted); (2) the text introducing and following the quote–the *context*–likely describes some facet of the quotation. The first observation says that there might be ways to distinguish ideas from other words; the second that commonly used descriptions may emerge without a commonly used classification system.

Furthermore, when one author quotes a passage from another they are crediting importance to the passage. In the aggregate this behavior can be seen as a "wisdom of crowds" [15] effect where authors' repeated re-use of one another's words call out popular and seminal passages. In other words, we found that mining surface similarity exposes deep semantics, i.e., the *ideas*, in books. The Key Ideas technique is based on these observations.

One major aspect of the emergence of popular and seminal ideas based on usage of quotations across books is the size and nature of the collection. The digital library in our inquiry is over a millions books that comprise Google Book Search[2]. This collection contains books from 28 libraries in the United States, Europe and Asia, content from over 10,000 publisher partners, and books in many dozens of languages.

1.1 Existing Quotation Collections

Quotation collections are fairly popular in print and on the Web. Two well known examples are Familiar Quotations by John Bartlett [4], and the Columbia World of Quotations [2] which contain 25,000 and 65,000 quotations respectively and are also available online[3]. Both these and other collections are generally well organized and excellent source books.

In developing our prototype we chose not to start from manually prepared quotation collections for a number of reasons. First, we were interested in quotations based on their "real world" use by authors and this is not the criteria used in edited collections. We also wanted as inclusive a collection as possible across many languages and disciplines. Finally, we wanted not just the quotation text, but the collection of books in which they are used as well as the context of use, something not available in existing online collections.

Further, searching from a large database of quotations is not a solution given the variations of use and numerous scan errors in our corpus. Our data mined collection of quoted passages contains tens of millions of entries, 3 orders of magnitude more than the largest edited collection. We know of no other quotation collection this comprehensive.

The remainder of this paper follows, starting with our motivation and a description of the overall user interaction experience. We then present algorithms for extracting key terms from quotation context and computing related key terms. We conclude with a broad discussion on what we learned building the prototype, what works, and where we see opportunities for further work.

2. MOTIVATION & RELATED WORK

This work falls within the research area of information seeking and, like much of the field, follows from Vannevar Bush's vision of Memex. Bush popularized the idea that people can collaboratively organize and share knowledge: "Wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails running through them..." [5]. In many ways the vision has come to pass with large amounts of hyper-linked information in the World Wide Web, digital libraries, and online encyclopedias. However, researchers continue to point out that people, especially those lacking domain expertise, struggle to navigate information spaces.

There are numerous reasons why navigating information is still a challenge. In our own encounters with a digital library of books we saw that the size of information chunks is fundamental. Following the "mesh of associative trails" and landing on a 400 page book slows down exploration. People like to see concise bits of information and then decide whether they want to learn more or navigate to other information.

2.1 Popular Passages

Our experience with Book Search led us to look for ways to connect readers to smaller, more interesting bits within books. We developed *Popular Passages* to help readers discover passages shared across books, and provide a way to explore the corpus by pivoting on these passages. The passages that authors decided worth repeating were mostly attributed quotations between 1-8 sentences in length, and occuring in tens, hundreds, or sometimes thousands of books. Mining quotations is related to but different from plagiarism and duplicate detection. For technical details on quotation mining see [9].

Although our system mines all types of repeated passages, our focus is not on shallow bon mots but rather on passages that authors re-use because they are relevant to a point being made and describe an idea well. These passages tend to be longer than a phrase but shorter than a page. For example, we find this passage frequently used in the field of environmental politics:

> The concept of sustainable development does imply limits - not absolute limits but limitations imposed by the present state of technology and social organization on environmental resources and by the ability of the biosphere to absorb the effects of human activities.
>
> – The Brundtland Report

As part of the process of identifying and mining repeated passages, we also link them together. Popular Passages may be compared to *tranclusion*, which Ted Nelson describes as: "what quotation, copying and cross-referencing merely attempt... Transclusions are not copies and they are not instances, but the same thing knowably and visibly in more than once place" [12]. Although the end result is similar, Nelson's transclusion comes about by creating documents

loquial expression or aphorism and a passage that is long is more likely part of a collection of works. In practice most of the "good" English language quotations we see are between 1-8 sentences. This follows more from the way that authors use quotation, rather than our algorithms.

[2] books.google.com

[3] See http://www.bartleby.com/quotations

Angel; Animal; Aristocracy; Art; Astronomy; Beauty; Being; Cause; Chance; Change; Citizen; Constitution; Courage; Custom and Convention; Definition; Democracy; Desire; Dialectic; Duty; Education; Element; Emotion; Eternity; Evolution; Experience; Family; Fate; Form; God; Good and Evil; Government; Habit; Happiness; History; Honor; Hypothesis; Idea; Immortality; Induction; Infinity; Judgment; Justice; Knowledge; Labor; Language; Law; Liberty; Life and Death; Logic; Love; Man; Mathematics; Matter; Mechanics; Medicine; Memory and Imagination; Metaphysics; Mind; Monarchy; Nature; Necessity and Contingency; Oligarchy; One and Many; Opinion; Opposition; Philosophy; Physics; Pleasure and Pain; Poetry; Principle; Progress; Prophecy; Prudence; Punishment; Quality; Quantity; Reasoning; Relation; Religion; Revolution; Rhetoric; Same and Other; Science; Sense; Sign and Symbol; Sin; Slavery; Soul; Space; State; Temperance; Theology; Time; Truth; Tyranny; Universal and Particular; Virtue and Vice; War and Peace; Wealth; Will; Wisdom; World

Figure 2: The 102 Great Ideas that comprise Encyclopedia Britannica's two volume index to the Great Books of the Western World. Each index entry contains a list of passages from books.

within an edit decision list (EDL) structure, whereas authors who write books don't generally use hypertext tools. In spite of these differences, the manufactured transclusions in Popular Passages, like those in Nelson's hypertext, land readers at a grounded point in the target document. This form of hyperlink alleviates the disorientation that occurs when following a link to the start of an entire book.

We developed Popular Passages to promote two types of navigation: within-book and between-books. First, while looking at a particular book a reader can navigate to "hot spots" within the text that have been popularly quoted. The idea is that this provides a quick overview or highlight view of the text. The second navigation technique we support is finding "similar" books by traversing a multi-way link from one popular passage to a list of other books containing that same passage. This link structure can provide a different analysis of a particular statement or historical event and can also let readers see the passage in the original source (if it is available in the library).

2.2 User Feedback

Our implementation of Popular Passages keeps the reader situated within books. Passages only exist when looking at books, and passages only link to books. When we showed this feature to users they asked for two additional features. First was the ability to search quotation text. The general statement was something like: "Can it show me that Al Gore 'sacrifice a tree for a human life' quote?"

The second type of comment we received was: "can it show me quotes about 'environmentalism'?" This second class of request was asking for a categorization of quotes by topic. Both of these questions pointed out the interest in treating quotations as first class entities, not just in the context of a book. This user feedback, particularly requests for features to explore by topic, motivated the design and implementation of the Key Ideas system reported in this paper.

2.3 The 102 Great Ideas

Exploring ideas across books by topic was a principal theme of the great books educational movement, and subsequent commercialization, in the latter half of the 20th century.

In 1952 the University of Chicago in collaboration with Encyclopedia Brittanica published the "Great Books of the Western World" consisting of 443 works in 54 volumes and a two volume index of the themes running through these books called the "Syntopicon" [1]. This index is the largest effort of its kind taking 7 years, involving, at its peak, 50 indexers, 75 clerical staff and costing over $1 million. The indexers created this "collection of topics" by examining over 900,000 passages for possible inclusion and maintaining a system on typed and re-typed index cards [3]. The organization of the index is 102 broad, top level themes (see Figure 2) and around 3,000 topic entries under these themes.

From the start the Great Books, and by association the Syntopicon, met criticism from progressive educators and even supporters of a great books pedagogical approach. In the New York Times Book Review from September 14, 1952, Gilbert Highet, a professor of Latin at Columbia University wrote that "In times like ours it is plainly impossible for 10 or 12 men, working in one single tradition, to select and to explain all the greatest books of the 3,000-year-old and 10,000-mile-wide West" [7].

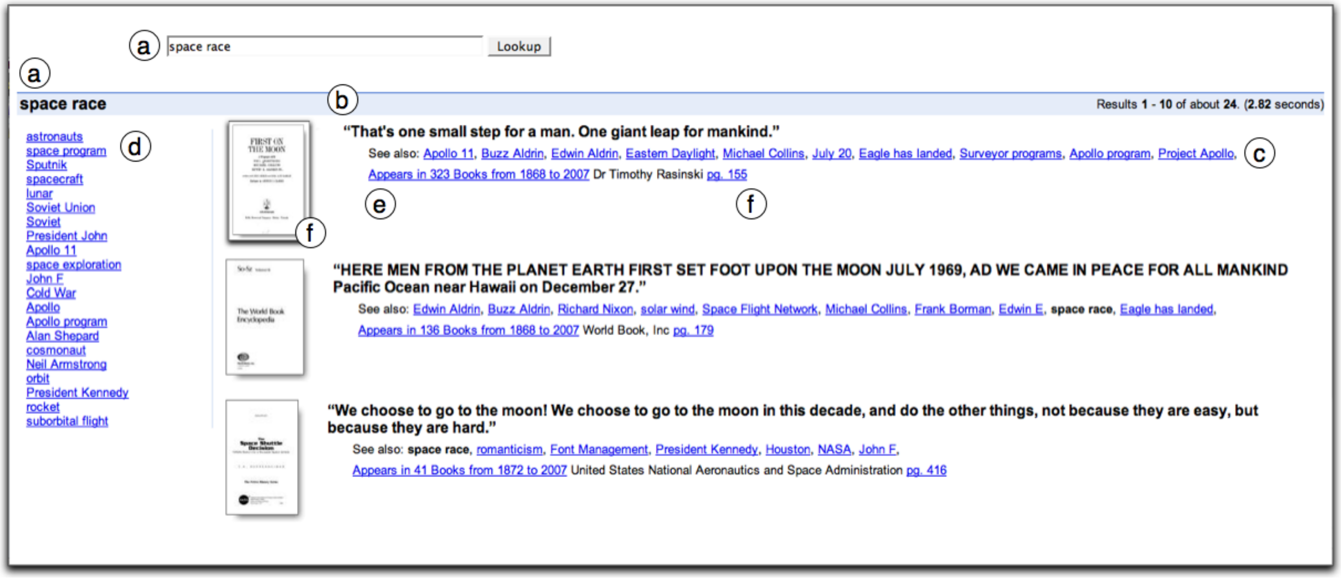
Others called the collection arbitrary and culturally biased towards white-males. Critics asked: why isn't Equality or Civil Rights a Great Idea; why is the Old and New Testament referenced but not the Koran? Some voiced concern that these Great Ideas are not "our" Great Ideas, a position supported by the lack of women and minority authors; it was not until 1990 that 4 women were added to 130 authors in the collection [11].

We were motivated to revisit an index of great ideas appearing across books–a syntopicon–and asked, is it possible to find great or simply good ideas without using an editorial board as agent? The following sections present the system we designed to explore this question.

3. KEY IDEAS INTERACTION MODEL

The interaction model for Key Ideas is built around three elements: books, quotations, and key terms. Key terms are generally 1-3 word phrases that include people, places, and things. Links exist between each element in both directions: quotations and books; quotations and key terms; key terms and related key terms. All links are one-to-many. Although the link structure sounds complicated, the main interaction cycle is easy to grasp: for the most part people click on key terms.

As a person clicks on key terms they see a view that looks similar to search results (Figure 3). The header shows the current key term and the result items are quotations that have been tagged with the term. Each result item also in-



space race

Results 1 - 10 of about 24. (2.82 seconds)

astronauts
space program
Sputnik
spacecraft
lunar
Soviet Union
Soviet
President John
Apollo 11
space exploration
John F
Cold War
Apollo
Apollo program
Alan Shepard
cosmonaut
Neil Armstrong
orbit
President Kennedy
rocket
suborbital flight

"That's one small step for a man. One giant leap for mankind."

See also: Apollo 11, Buzz Aldrin, Edwin Aldrin, Eastern Daylight, Michael Collins, July 20, Eagle has landed, Surveyor programs, Apollo program, Project Apollo,

Appears in 323 Books from 1868 to 2007 Dr Timothy Rasinski pg. 155

"HERE MEN FROM THE PLANET EARTH FIRST SET FOOT UPON THE MOON JULY 1969, AD WE CAME IN PEACE FOR ALL MANKIND Pacific Ocean near Hawaii on December 27."

See also: Edwin Aldrin, Buzz Aldrin, Richard Nixon, solar wind, Space Flight Network, Michael Collins, Frank Borman, Edwin E, **space race**, Eagle has landed,

Appears in 136 Books from 1868 to 2007 World Book, Inc pg. 179

"We choose to go to the moon! We choose to go to the moon in this decade, and do the other things, not because they are easy, but because they are hard."

See also: **space race**, romanticism, Font Management, President Kennedy, Houston, NASA, John F,

Appears in 41 Books from 1872 to 2007 United States National Aeronautics and Space Administration pg. 416

Figure 3: The Key Ideas display consists of (a) key term entered manually or by navigation; (b) a list of quotations labeled with the term; (c) key terms associated with each quotation; (d) related key terms; (e) links to all the books containing the quotation; (f) link to a "default" book containing a full version of the quotation.

cludes a book cover and other elements which are described later. Essentially people view a synoptic layer on top of the book corpus that provides a general summary of the key term. So, for example, if the key term were "classical liberalism" they might see passages with a famous quote, a popular definition, a criticism, a comparison. For each key term there are often hundreds of quotations over many results pages, so we present them ranked by popularity and length.

There are two types of key terms users can click on. First, below each quotation is a list of the most relevant terms associated with the quotation. Second, on the left margin is a list of the most relevant terms associated with the view.

Key terms below a quotation are based on the instances of a single quotation so they tend to specific. The terms on the left margin are computed using frequently co-occurring terms across all quotations so they tend to reflect more general relations. For example *cold war* might appear in the left margin when viewing *space race* but not necessarily appear in all quotations. And in contrast, *Dan Quayle* and *Jack Kennedy* might appear together in quotations about the 1988 Vice-Presidential Debate, but not on the left margin when viewing either of these terms.

Clicking on either type of key term has the same effect of shifting the view to display quotations labeled with the new term, so we feel it is not necessary that people understand exactly how these are computed, but rather that they see a good selection of terms. We describe the algorithms for computing each type of key terms in section 4 and 5.

In summary, the main interaction cycle involves clicking on key terms and reading quotations. The effect is like skimming through a layer on top of book content. The cycle is entered by a user entering a key term in a text box or clicking on a key term from another part of our web site, for example when they are looking at a book. When a user finds an interesting quotation they can either dive down and explore source books or copy the passage to a notebook. These are described below.

3.1 Diving into Books

The Key Ideas interface includes two ways to explore books when looking at quotations. First, people can click on the link that says "Appears in *n* books..." and see a list of all the books where the quotation appears. They can then click on one of the book links and see the quotation in context on a page in the book.

Since people have a two step process to traverse this multiway link we thought it would also be useful to show a "default" link that jumps directly into a book. By clicking on one of the book images in Figure 3, the user quickly sees the quotation in context on a book page. Our initial thought was that the default book should be the earliest published instance, which would likely be the primary source.

Unfortunately we found that the earliest published book is often not the primary source. Sometimes the primary source was reissued with a later publication date and the original doesn't exist in the library, or another book's publication date is incorrect.

We developed a technique to better approximate primary source by finding which instances have the maximum length version of the quotation. We say these contain the *longest-quote*. The idea is that if a primary source exists then it too contains the longest-quote. Once we have this set, we take the instance with the earliest publication date for the default. The next sections describe this algorithm.

Longest Quotation Text
Although we may never know with complete certainty the identity of the winner of this year's presidential election, the identity of the loser is perfectly clear. It is the nation's confidence in the judge as an impartial guardian of the law.

Instance Text with Pre and Post Context	
1.	As Justice John Paul Stevens wrote in a blistering dissent, "**Although we may never know... law**" And that confidence may be further shaken if independent full manual recounts by the press (allowed under Florida's sunshine laws) decide that Mr Gore did indeed "win"
2.	It chastised the Court for holding the justices of the Florida Supreme Court up to ridicule. "**Although we may never know... law**"
3	"**The identity of the loser is perfectly clear**" wrote Justice Stevens, "**It is... law.**"
	...
81.	On December 13, Al Gore conceded the election to George W. Bush. The high court's ruling was uncharacteristically long-winded and excruciatingly tortured in its logic. Dissenting from the majority, Associate Justice Stephen Breyer wrote that while "**we may never know...**"

Figure 4: Justice Steven's opinion from the *Bush v. Gore* Supreme Court Case No. 000-949 was quoted in 81 books in our corpus. Our algorithm extracts key terms from the context of each quotation. Specifically a few hundred characters before and after each instance of the quotation are appended to form a document. This document is then run through a key term extractor. The key terms that emerge usually include the author, the subject, and related names and terms, such as "manual recount." All key terms are extracted statistically to allow for the dozens of languages in the collection. (Ellipses added for readability).

3.1.1 Longest Quote Instance

Our system groups together instances with different length text as long as they have a common run of words above a threshold. For example, there are 81 instances (books) with the quotation by Justice Stevens in Figure 4. Most of these are the 2-sentence version shown. However, the author for instance 3 splits the text into parts, and the author for instance 81 starts the quote in mid-sentence. The technique for grouping is described elsewhere [9], but it is sufficient to know that various lengths of a quote are grouped. Also, in practice many of these instances have different lengths because OCR errors will break long matches.

Computing the *longest-quote set* involves processing n instances. We start by setting d_j, $1 < j < n$, to the concatenation of quote text and context for each instance. Let $S(d_j)$ be the k-shingling[4] of text d_j. We then compute the global occurrence count $O(s_i)$ for each shingle s_i over all $S(d_j)$ using a hash table. We then go over $S(d_j)$ and count the number of shingles for which $O(s_i) > 1$, and pick the instances that has the maximum count. More formally, we are after $\text{argmax}_i(|\{s|s \in S(d_i) \wedge O(s) > 1\}|)$ and we can have multiple i that satisfy this. This set of instances then represents the longest-quote set.

This technique works reasonably well given OCR errors and variations in author quotation style. An alternative would be to compute greatest common-substring with errors using suffix trees, however our system already makes extensive use of shingling. Note that the longest-quote is not necessarily the quote that is displayed, instead we use a statistical algorithm to find the most frequent form of the quotation for display purposes (these details are described in [9]).

Linking to all books and linking to a default book are two ways we support the activity of diving into and exploring the content inside books. When users find an interest in a particular quotation they can easily jump from the quotation into a book page, and read the original author's words in context, or read analysis, criticism, or discussion by other authors. The next activity we wanted to support was collecting information.

3.2 Note Taking

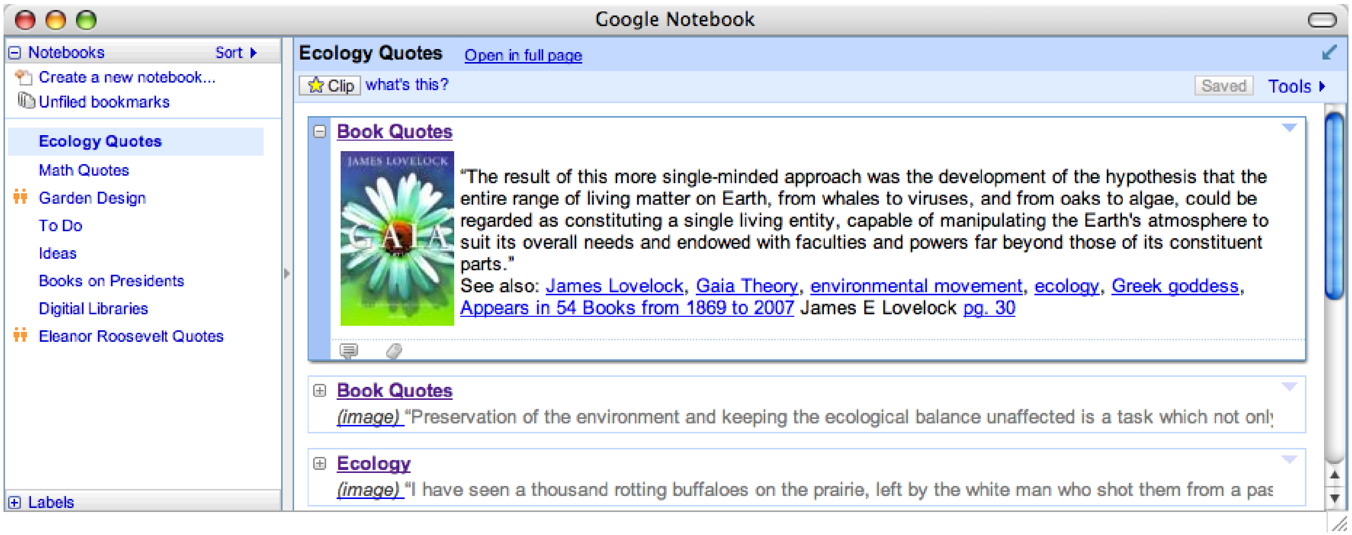
People exploring information will commonly also engage in note taking. We considered building this feature into our browser, but found an existing tool, Google Notebook, supports note taking especially well (see Figure 5). While a person is exploring quotations they can select and copy a result item into Notebook using the "Note this" context menu. Result items in notebook maintain the same look and active links as those listed on the Key Ideas page. Result items can be saved in different notebooks or notebook sections and tagged with labels. This makes it easy to create and organize collections. Quotation collections stored in notebook can also be exported or shared with others.

4. LABELING WITH KEY TERMS

As described in the previous section, interactions in our system are driven by users clicking on key terms such as: "mass media," "groupthink," "Stanley Milgram" or "social psychology." We could generate this set of labels by running a key term extractor over the quotation text. However, quotations themselves do not necessarily include the most descriptive terms and moreover, one author writing on the "boomerang effect" might be covering the same idea as "psychological reactance" and these terms would not be connected.

Our approach therefore is to extract key terms from the descriptive context surrounding all instances of a quotation text. We concatenate a short context from each occurrence of the passage and run a key term extractor over the resulting string as if it were a document. Since authors often formally introduce quotations to their readers, the quote author's name is usually in the context. Likewise because

[4] A k-shingling of d is the set of all consecutive k-grams in d.



"The result of this more single-minded approach was the development of the hypothesis that the entire range of living matter on Earth, from whales to viruses, and from oaks to algae, could be regarded as constituting a single living entity, capable of manipulating the Earth's atmosphere to suit its overall needs and endowed with faculties and powers far beyond those of its constituent parts."

See also: James Lovelock, Gaia Theory, environmental movement, ecology, Greek goddess, Appears in 54 Books from 1869 to 2007 James E Lovelock pg. 30

(image) "Preservation of the environment and keeping the ecological balance unaffected is a task which not onl

(image) "I have seen a thousand rotting buffaloes on the prairie, left by the white man who shot them from a pas

Figure 5: Note-taking is supported by integration with Google Notebook where quotes can be selected and copied into tabbed sections.

quoted passages are used to support an author's idea, associated ideas are in the context. If a quotation is used to support new ideas or ideas with different terms, these would also be reflected in the context.

A novel aspect of our system is that all authors who use a quotation collectively assist in classifying it. This "folksonomy" addresses the problem of finding a common classification system in the face of synonyms and evolving terminology. We use existing key term extraction methods because these too are designed to weight key terms by their frequency across uses. We could also search the context for key terms appearing in a database and apply frequency weights separately. However, key term databases covering the range of languages in the corpus are not readily available.

4.1 Labels versus Authors

We initially thought it would be useful to separate the speaker of the quotation from other names and key terms. In this case we might have a page for quotes attributed to John Paul Stevens, rather than the broader categorization that includes quotes by and about Stevens.

It turns out that attribution is not so simple. For example, early on we found that many quotes are attributed to multiple people, including this: "I read somewhere that everybody on this planet is separated by only six other people." These words are spoken by Ouisa Kitteridge, a character in the play *Six Degrees of Separation* by John Guare. People refer to the quote as coming from both fictional character and the playright. We were also reluctant to parse speech-acts or other language specific features. In general we prefer statistical over language specific methods in order to equally support the dozens of languages in the collection. In the end we decided to avoid these issues and adopt the style of user contributed labels seen on photo and bookmark sharing sites.

Figure 4 provides an example where the quoted passage is from Justice Steven's opinion in the *Bush v. Gore* United States Supreme Court case. In our analysis this passage appears in 81 books. In each of these 81 books the passage fits in a flow of author's words that include key terms such as "Al Gore," "manual recounts" and "Florida's sunshine laws."

After examining the passage for this example we noticed that 3 of the books attribute the passage to Associate Justice Breyer. Although the dissenting opinion was signed by Breyer and Ginsburg, it was written by John Paul Stevens. We think this example supports the decision to use labels with weights over extracted speaker names.

4.2 Key Term Extraction

Our key term extractor starts by generating word n-grams for size 1-4 for the concatenated context document. The algorithm then discards n-grams that start or end on any stop-word found in a statistical per-language stop-word tables. The next phase converts to lower-case, applies Porter stemming if available, collapses similar terms, and records the term frequencies. During this phase we also store the most frequent surface form counts for later display. The algorithm then looks up document frequency for the n-gram in a pre-computed language specific table generated for our corpus. We also look up a "key phraseness" value for the n-gram in a table of known key phrases. The term frequency, document frequency, and key phraseness values are combined to produce a key term value. The n highest scoring key terms are associated with each quotation using the most popular surface form for the display label.

One surprise we had about context is that there is a lot of it. Many quotations appear in tens or hundreds of books in our corpus. The somewhat banal passage about sustainable development (see Section 2.1) appears in 51 books. This means that if we use a couple of sentences before and after each quote as context, we have over 200 sentences that can be used to label the quotation.

The substantial challenge we faced in key term extraction was coming to an understanding of what "context" means, and then being able to collect *clean context*. It turns out that *dirty context* breaks statistical key term extractors. We discuss these two issues next.

4.3 What is Context?

Under our current processing pipeline, we found it is practical to include only a few hundred characters before and after a passage as context for key term extraction. We use a fixed number of bytes rather than words or sentences to facilitate processing. If the cutoff falls within a word, we adjust to a word boundary. We postulated that a couple of sentences worth of text would generally capture attribution and concept labels. If this was not the case in some specific instance, then it would still be the case statistically over all context. We also postulated that there is a diminishing returns with larger context, and in our situation the overhead of aggregating large amounts is substantial. We have left the study of performance of key term extraction from paragraph, page, chapter or even book size context to future work.

One may also ask whether "context" exists for the original source. Some of the books in the example contain the full Supreme Court opinion, the original source, where other books contain just a few sentences. Should the full opinion be included in context computation? We found that it is not necessary to separately detect and process primary source instances. Although it is true that secondary sources are different, for example that author's will tend to introduce the passage with attribution, in general both provide reasonable descriptive context.

4.4 Collecting Context

Our system needs to carefully distinguish quotation text from context. To see why, consider the author of instance 3 in figure 4 who splits the quote into parts, interposing "Stevens wrote." If the first part of the quote is below our threshold length for detecting repeating passage text, then it will appear as context. Consequently, context n-grams like "perfectly clear" will end up with a boosted term frequency because they also occur in the quotation text. The result is that word n-grams that are not actually good key terms start appearing in our labels.

Pieces of the quotation text appearing in context also regularly occur because OCR errors will split a quotation. One solution is for our earlier processing phases to do a better job of approximate string matching, however this is a relatively expensive task at that point.

We tried a number approaches to solve this problem. We arrived at a technique of taking all the text and context in all instances of the quotation and reprocessing them. We start by computing the longest-quote instance described in section 3.1.1. This represents an instance with the largest number of shingles appearing in other instances.

The algorithm re-processes all text and context for n instances. We let $N(d_f)$ be the n-grams of the longest-quote instance, and $N(d_j)$ be the n-grams of instance j. As we process n-grams for key term extraction we skip n-grams in $N(d_f) \cap N(d_j)$. This is strict but the key term extractor also processes a copy of the longest-quote instance, so good key term candidates in full-quotation whose term frequencies are being suppressed will generally get sufficient boost when key phraseness is applied.

5. RELATED KEY TERMS

The key term extraction step is used to associate a set of key terms with each quotation. However, the relations between key terms remains implicit. For example, just by looking at a few quotations under the term "space race" it is not easy to see a relation between the "space race" and the "cold war." However, if the reader inspected all of the quotations this relation would appear.

While it is possible for people to discover the relations between key terms manually, providing related key terms as part of the user interface simplifies the process.

5.1 Extraction Method

Our related key term extraction method is based on the well-known idea of co-occurrence. If two key terms appear in quotations more often than possible by chance, they are probably related to each other. There are various metrics that can be used to measure level of relatedness based on co-occurrence, such as inner product, normalized co-occurence, Dice's coefficient, Cosine similarity, Jaccard coefficient [8], log-odds, and mutual information [14]. Since this is a well-studied area, we will not focus on the specific measure to use, but rather describe how we collect the statistics necessary to compute these similarity measures and perform the computation in an scalable way.

For two key terms k_1 and k_2, if we define quotation sets S_1 and S_2 such that $S_1 \equiv \{s \mid k_1 \text{ is associated with } s\}$ and $S_2 \equiv \{s \mid k_2 \text{ is associated with } s\}$, all we need to know to compute any of these similarity mesures between k_1 and k_2 are $|S_1|$, $|S_2|$, $|S_1 \cup S_2|$, $|S_1 \cap S_2|$, and $|S|$.

One way to compute the required statistics is to create an index for each key term that holds the set of quotations that are associated with the key term. This is like the inverted indices used by search engines to map key terms to documents. We can collect statistics required for similarity computations by doing lookups into this index. However, since we need to compare every key term to every other key term, this involves many lookups which is feasible for small digital libraries but impractical for large collections. Instead, we make use of the MapReduce (MR) paradigm [6] to perform a massively parallel batch processing which is pretty much brute force except we never consider a pair of key terms for relatedness if they were never associated with a common passage.

For simplicity, we will use k_1, k_2, and their associated sequence sets to describe our related key term computation. While describing the input and output of map and reduce operations, we will use the notation $[key : value]$ where key, value, or both can be a tuple of the form $\langle item_1, item_2, ...\rangle$.

The input for the map phase is the quotation database, which contains all the quotations we extracted along with their associated key terms. For each map operation, we take one quotation with its key terms, and output each key term once, and all possible pairs of key terms once, with a value of 1. For example, for a quotation p_1 with key terms $\{k_1, k_2, k_3\}$, we would output $[k_1 : 1]$, $[k_2 : 1]$, $[k_3 : 1]$, $[\langle k_1, k_2\rangle : 1]$, $[\langle k_1, k_3\rangle : 1]$, $[\langle k_2, k_3\rangle : 1]$.

In order to make sure the pair has a unique representation across all map operations, the key terms are listed in alphabetical order. The reduce phase is a simple sum reducer, which counts how many quotations are associated with each key term and key term pair. The final result is a map from key terms and pairs of key terms to their counts. So for

k_1 and k_2, we know $|S_1|$, $|S_2|$, and $|S_1 \cap S_2|$. $|S_1 \cup S_2|$ can be dervied from these. We know the number of quotations, $|S|$, beforehand, and even if we did not it could be trivially obtained using the MR that we just described.

The next step is to compute the association metric of choice for every pair of key terms that ever co-occured in the passage collection. We can compute all the similarity metrics for key term pair $\langle k_1, k_2 \rangle$ by doing three lookups into the table generated above, for $|S_1|$, $|S_2|$, and $|S_1 \cap S_2|$.

However, instead of doing N^2 lookups and computations over the statistics table we have generated, we run another MR. We started with the inner product metric in our experiments, because all we need in order to compute this metric is $|S_1 \cap S_2|$, and nothing else. This simplifies the implementation and reduces the amount of data transfer and execution time, allowing faster experimentation. Therefore, we will first describe the MR that we use to compute the inner product, followed by the design of the MR that can be used to compute the other metrics.

5.1.1 Inner Product MapReduce

Input to this MR is the statistics table generated previously, and the output is a table that maps each key term to all related key terms along with the relatedness score. For each map operation, the input is either a key term or a key term pair and its count, specifically, we will get $[k_1 : |S_1|]$, $[k_2 : |S_2|]$, $[\langle k_1, k_2 \rangle : |S_1 \cap S_2|]$.

We ignore the single terms as they are not needed for the inner product computatuion. For pairs, we produce one output for each key term in the pair. The output key is the term and the value is the other term along with the count for the pair. For $[\langle k_1, k_2 \rangle : |S_1 \cap S_2|]$, the outputs would be $[k_1 : \langle k_2, |S_1 \cap S_2|] \rangle$ and $[k_2 : \langle k_1, |S_1 \cap S_2| \rangle]$.

In the reduce phase, for each reduce call the key will be a particular key term and values will be all the key term pairs and their counts where this term appears. For our running example, we would get a reduce call for key k_1 with the value $\langle k_2, |S_1 \cap S_2| \rangle$. This is enough information to compute the inner product based relatedness between the key term and all the other key terms with which it co-occured. We can then filter key terms with low relatedness scores if we like, and output the remaining key terms and their scores as the related key term set for the current key term.

5.1.2 MapReduce for Other Similarity Measures

Our goal is to be able to compute all the similarity measures mentioned above for any given key term. The MR described above does not allow this because reducer has access to the counts for the pairs, but not the counts for individual key terms involved in those pairs. Our new design requires two MRs. The first one identifies all the pairs a key term is involved in, grouping the key term, its count, and all its pairs together. The second one takes these groups, distributes the individual term counts to all the pairs involved, and computes the similarity metric for each key term pair.

Let us start with the first MR. Just like the MR we described above, for each map call, we either get a single key term or a pair, and its count. For key terms, $[k_1 : |S_1|]$, we simply copy the input to output. For key term pairs, the output is identical to the MR described above. In the reduce phase, each reduce call for a given key term will get the individual count for that key term, and all the pairs that key term appears in and their counts. For example, the reduce call for k_1 will get the values $|S_1|$ and $\langle k_2, |S_1 \cap S_2| \rangle$. We group all the values for a given key, and write them out. For k_1, the output would be $[k_1 : \big\langle |S_1|, \langle k_2, |S_1 \cap S_2| \rangle \big\rangle]$. At this point, for k_1, we have all the values we need except $|S_2|$. We use a second MR to get the missing piece.

The input for the second MR is the table generated by the first one. For each map call, we get a key term, its count, all the pairs it appears in, and their counts, e.g. $[k_1 : \big\langle |S_1|, \langle k_2, |S_1 \cap S_2| \rangle \big\rangle]$. We generate one output for the key term and its count, $[k_1 : |S_1|]$. For each pair $\langle k_1, k_2 \rangle$ in the list, we produce two outputs: $[k_1 : \langle k_2, |S_1 \cap S_2| \rangle]$, $[k_2 : \langle k_1, |S_1| \rangle]$.

At the reduce phase, reduce will be called once for each key term. For k_1, we will get $|S_1|$ and $\langle k_2, |S_1 \cap S_2| \rangle$ as values as they were output by the map call for k_1. We also get $\langle k_2, |S_2| \rangle$ as a value, since it would be output by the map call for k_2. Using these values, we can compute $|S_1 \cap S_2|$, which completes the data set necessary for similarity computations. We can now compute the similarity metric of our choice and genereate the list of related keyterms.

6. DISCUSSION

We set out to design and build a system that lets people explore the growing library of online books. In this section we discuss where we think we have been successful, where we have failed, and areas where we see progress can be made.

First, we think the interaction model is a success. The ability to skim short, and mostly interesting, quotations; pivot to related quotations; and dig into book pages where the quotations are used is engaging. Indeed, on our product site Popular Passages' book-centric feature is one of the most used navigational methods [9]. We are surprised at how many passages have been quoted and how broadly many of them have been used. We are also pleased that terms like "space race" appear with a reasonable set of quotations and that related-terms produce a rich network of labels connecting ideas (See Figure 6.).

6.1 Intelligence versus Wisdom

Although we set out with a comparison with online encyclopedia, its clear these are two deeply different collaborative activities. Wikipedia is an activity of collective intelligence, where diverse groups pool knowledge, collaborate and write together. Key Ideas is an activity of crowd wisdom. Surowiecki describes the four key qualities that make a crowd smart as:

> It needs to be diverse, so that people are bringing different pieces of information to the table. It needs to be decentralized, so that no one at the top is dictating the crowd's answer. It needs a way of summarizing people's opinions into one collective verdict. And the people in the crowd need to be independent, so that they pay attention mostly to their own information, and not worrying about what everyone around them thinks.

We think we have demonstrated that a *wisdom of the crowds* effect is present when we mine and count quotations in books. However, we don't think this system, in its current form, can replace one based on collective intelligence. In our system you tend to find and learn new ideas without much context for understanding how they fit into a broader field. There is a large gap between reading popular quotations and

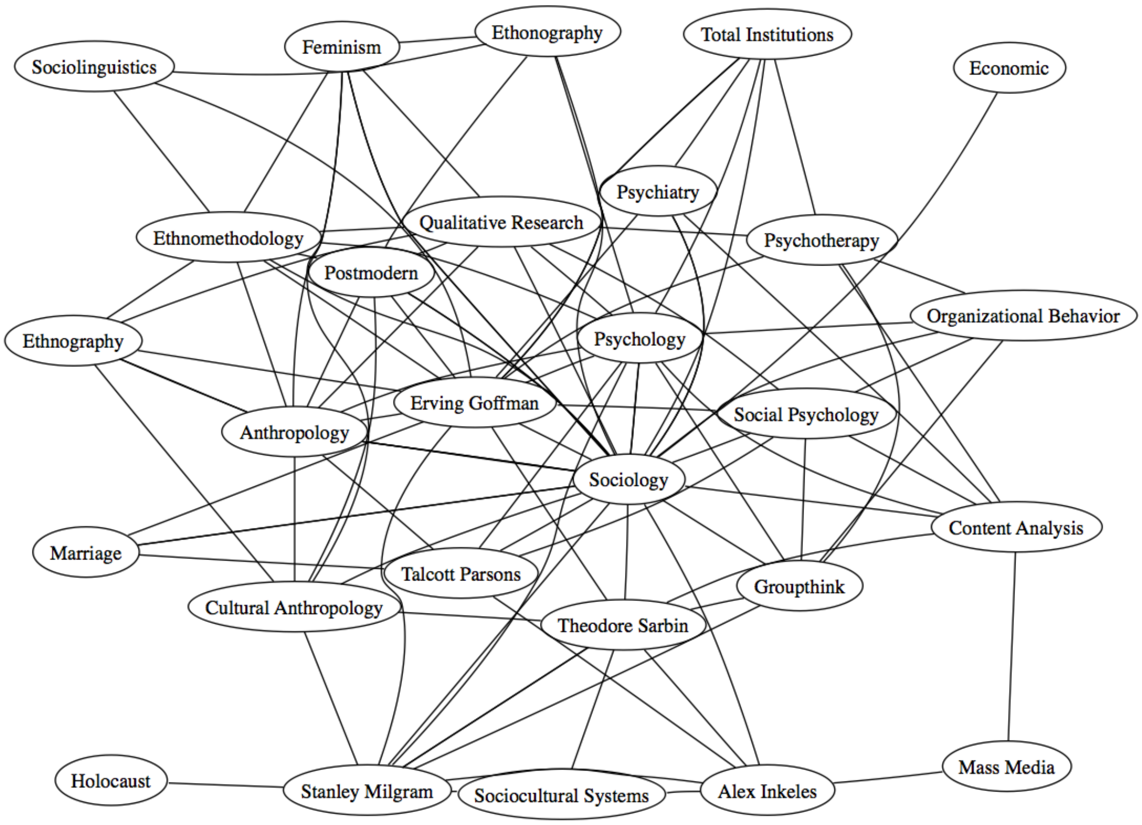


Figure 6: A small portion of the "idea graph" generated from tens of million quotations mined from books.google.com. Each node is labeled with a key term extracted from the context words around a quotation as it appears in multiple books. Each edge denotes key terms that co-occur among multiple quotations.

reading and understanding book pages. What is missing is the human edited summaries.

6.2 Exploration versus Seeking

Although we feel success providing a browsing and exploration experience we fail to provide a good investigation and information seeking experience around quotations. Clearly full text search of quotation text is needed.

From the interaction side one issue that comes up from users is that key terms can't be combined. Sometimes users click on a term expecting a smaller set but they get a larger set: "Rosalin" when reading on "Jimmy Carter." Also synonyms are not particularly well handled: "President Carter" and "Jimmy Carter" often appear side by side.

6.3 Authorities and Hubs

Another useful area to pursue is giving readers a better sense of authorities and hubs. On one hand when presenting all the quotes for a key term our algorithms should boost quotations from authoritative sources. On the other hand when presenting all the books containing a particular quotation, our algorithms should boost the authoritative texts containing that quote. Recognizing that author's of insightful analysis are themselves quoted leads to a network, reminiscent of page-rank, that would be interesting to expose and exploit.

In this area of investigation, it would also be useful for the system to give viewers a sense of the use of ideas over time and place. How have ideas caught on or evolved? Which ideas lost popularity after the demise of the Ottoman Empire and how long did it take Charles Darwin's ideas about evolution to influence various cultures?

6.4 Collection Facets and Bias

Another shortcoming of our work is that we don't distinguish between fiction and non-fiction. It might also be useful to let readers limit their exploration by subject categories, such as "linguistics." Since online books come from hundreds of highly managed collections it may also be useful to look at facets by collection. For example, popular quotations from the Bayerische Staatsbibliothek (Bavarian State Library).

The larger issue is that there is no control of the bias that the corpus introduces. We saw that Brittanica's Syntopicon, although a successful index of concepts, was criticized because it was based on a selection of books by "10 or 12 men, working in one single tradition." We believe that when collections grow to millions of books some issues are reduced. However, one avenue of further research is how might we let readers positively control bias by manipulating the underlying corpus of books used for extraction.

7. CONCLUSION

Our work is motivated by the vast amounts of digitized book content that is now appearing online. This infusion of content has created a text-rich but hypertext poor region of the web. Although we are seeing the introduction of citation and other types of links, these tend to target the entire book. Our development of Popular Passages addressed the link target issue. However, people were also interested in viewing information outside the constraints of a book's bindings, and exploring information by topic across books.

This paper described Key Ideas, a new exploration technique that lets people browse frequently quoted passages from online books, pivot to related quotations, and dive into source books to read the quotations in context. Our technique organizes the browsing experience around key terms extracted from the context of quotations.

We created our quotation database and associated key terms from the online books in Google's Book Search library, consisting of over a million scanned books in dozens of languages. We believe the resulting quotation database with tens of millions of entries is the largest of its kind.

Our system exposes the discussions that have been going on between authors since publishing began. By counting how often quotations are used across books we extract and uncover the noteworthy and seminal statements that have been copied from author to author.

The crowd wisdom of Key Ideas complements the collective intelligence of other tools. From Key Ideas people can skim ideas and read deeper in source books, from Wikipedia people get an overview of how these ideas fit together. From our user experience, Key Ideas is a promising new way to connect people to the knowledge that lies buried in massive collections of online digital books.

8. ACKNOWLEDGMENTS

We'd like to thank our colleagues who provided ideas, feedback, and encouragement during this research. Adam Mathes has been a long time contributor to discussions as well as UI designs. Kathrin Paschen contributed to work on primary source identification. Andras Csomai built the basic key term extractor. We appreciate the hard working students who helped us explore a number of technical issues: Justin Vincent, Andrew Gadson, Ryan Fortune.

9. REFERENCES

[1] M. J. Adler, C. Fadiman, and P. W. Goetz, editors. *The Syntopicon: An Index to the Great Ideas.* Encyclopedia Britannica, 2nd edition, 1990.

[2] R. Andrews, M. Biggs, M. Seidel, and et al., editors. *The Columbia world of quotations.* Columbia University Press, New York, 1996.

[3] Anonymous. Fusilier. *Time Magazine*, March 17, 1952.

[4] J. Bartlett. *Bartlett's Familiar Quotations: A Collection of Passages, Phrases, and Proverbs Traced to Their Sources in Ancient and Modern Literature.* Little, Brown and Company, 17th edition, 2002.

[5] V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, July 194.

[6] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[7] G. Highet. Ideas that shape the minds of men; great books of the western world. *The New York Times, Book Review*, page BR1, September 14 1952.

[8] P. Jaccard. Le coefficient générique et le coefficient de communauté dans la flore marocaine, 1926.

[9] O. Kolak and B. N. Schilit. Generating links by mining quotations. In *Proceedings of the nineteenth ACM conference on Hypertext and Hypermedia (Pittsburgh, Pennsylvania, United States, June 19-21, 2008). HYPERTEXT '08*, New York, NY., June 2008. ACM.

[10] G. P. Landow, editor. *Hypertext: The Convergence of Contemporary Critical Theory and Technology.* The Johns Hopkins University Press, Baltimore and London, 1991.

[11] E. McDowell. 'Great Books' Takes In Moderns and Women. *Time Magazine*, October 25 1990.

[12] T. H. Nelson. Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Computing Surveys*, 31(4), December 1999.

[13] K. Q. Seeyle. Snared in the web of a wikipedia liar. *New York Times*, December 4 2005.

[14] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3,4):379–423,623–656, July, October 1948.

[15] J. Surowiecki. *The Wisdom of Crowds.* Anchor Books, August 2005.