BIT Numer Math (2011) 51: 127–154
DOI 10.1007/s10543-011-0324-6

BIT

Some computational tools for digital archive and metadata maintenance

David F. Gleich · Ying Wang · Xiangrui Meng · Farnaz Ronaghi · Margot Gerritsen · Amin Saberi

Received: 13 September 2010 / Accepted: 22 February 2011 / Published online: 11 March 2011

Abstract Computational tools are a mainstay of current search and recommendation technology. But modern digital archives are astonishingly diverse collections of older digitized material and newer "born digital" content. Finding interesting material in these archives is still challenging. The material often lacks appropriate annotation—or metadata—so that people can find the most interesting material. We describe four computational tools we developed to aid in the processing and maintenance of large

Communicated by Axel Ruhe.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04- 94AL85000.

The majority of David's work was completed while at Stanford University.

D.F. Gleich
Sandia National Laboratories, Livermore, CA 94550, USA
e-mail: dfgleic@sandia.gov

Y. Wang · X. Meng · M. Gerritsen (✉)
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA
e-mail: gerritsen@stanford.edu

Y. Wang
e-mail: yw1984@stanford.edu

X. Meng
e-mail: mengxr@stanford.edu

F. Ronaghi · A. Saberi
Management Science and Engineering, Stanford University, Stanford, CA 94305, USA

F. Ronaghi
e-mail: farnaaz@stanford.edu

A. Saberi
e-mail: saberi@stanford.edu

digital archives. The first is an improvement to a graph layout algorithm for graphs with hundreds of thousands of nodes. The second is a new algorithm for matching databases with links among the objects, also known as a network alignment problem. The third is an optimization heuristic to disambiguate a set of geographic references in a book. And the fourth is a technique to automatically generate a title from a description.

Keywords Graph layout · Metadata remediation · Dynamic programming · Network alignment

Mathematics Subject Classification (2000) 05C50 · 05C85 · 68T50 · 90C39

1 Introduction

Around 1994, the Library of Congress engaged in a massive digitization effort of some of America's most culturally significant materials. The result was a collection called *American Memory* with a website interface. Among the digitized collections are George Washington's diary, Abraham Lincoln's letters, and the first films recorded by Thomas Edison. Getting people to the material in this collection, however, has been difficult. Although some limited metadata was collected during the initial digitization, the focus of those efforts *was* the digitization instead of robust access to the materials. Almost 20 years later, the Library is interested in updating these collections to modern digital archive standards. At a minimum, this requires accurate metadata on subject, place, time, and people.

Historically, librarians or subject matter experts created this metadata. Given the current ease of digitization, however, the quantity of materials has rapidly outpaced the ability of experts to annotate it. UNESCO recently launched the World Digital Library,[1] an attempt to put the most significant artifacts from the world into an online digital archive. The size of the initial collection was limited because of the desire for properly curated metadata, manually translated into each of the seven UN languages. Should our access to these artifacts be restricted by the onerous task of expert annotation and translation?

Let us summarize the problems with building search and browse tools in these archives. First, the items are highly heterogeneous—books are only a small portion of the searchable collection. Second, the metadata for everything except for books is inconsistent or incomplete, and the most useful metadata may not be available. Third, there are no native links between items. Fourth, the content spans many languages. Fifth, ranking these items in light of highly inconsistent metadata is challenging. For more background on these challenges, please see Sect. 2. We phrase our high level vision in terms of a virtual librarian service, which we view as a motivating ideal for future library search systems.

This paper does not present a comprehensive solution to these problems. Instead, we extract small, tractable, and interesting computational problems.

[1] See http://en.wikipedia.org/wiki/World_Digital_Library for more information.

The first problem we discuss is the jumble of data available. See Table 2 for an overview of the data we want to search and the data we can use to help search. We describe each of these datasets more completely in Sect. 3. While our overarching goal is to enable a unified search and browse interface, the objects that we want to search and browse are diverse. Another problem is that some of the information that we may wish to use is quite complicated. For example, the Library of Congress subject headings is a thesaurus used to uniquely identify subject matter. It has been around for over 100 years. There are entire courses in information science curricula about this database. How then can we quickly learn about it? Our answer is visualization, and we elaborate on this approach in Sect. 3.

Once we understood the structure of the Library of Congress subject headings, we noticed it was related to the structure of the categories underlying Wikipedia. This led to an exploration of how we could *match* the subject headings in the Library of Congress subject headings with the categories in Wikipedia. And moreover, it led us to consider using other sources of *open* or *crowd-sourced* data. Section 4 discusses our idea to match the Library of Congress subject headings with Wikipedia categories, and it also discusses challenges in using these types of data.

At this point, we arrive at a key problem. The most desirable data is about the *place* and *subject* associated with each object in our collection. However, this data is not always available. The next two sections propose ideas to generate this *missing metadata*. In Sect. 5, we introduce an optimization problem to disambiguate references to places and events. Among other uses, it tries to answer the question: does "San Jose" refer to San Jose, California or San Jose, Costa Rica? In Sect. 6 we describe how to automatically generate a title and a set of keywords from a textual description.

These last three problems we described are about *generating* new information. While these endeavors are helpful, we also need to ascertain the quality of this new information. Evaluating any information system is an arduous affair because *only human responses* can be considered the ground-truth. Section 7 describes our evaluation efforts. We conclude with a wide range of interesting problems for future research (Sect. 8.2).

2 Background

Over the past 25 years, we have witnessed a shift in the nature of our society's records. Previously, these records were stored on paper or an alternative physical media. Now, records are usually digital files. This situation poses a subtle problem. Consider how much of your own work—digitally preserved—is no longer accessible because:

- the program to read the file is no longer available;
- the program to read the file no longer works with old files;
- there is no longer any hardware to read the physical media.

Kuny [29] lays out the basis for the problem and coined the phrase *a digital dark age* to convey the gravity of the situation. He also describes some of the solutions needed to address it. Mostly, these ideas focus on the problem of preserving the digital bits, storage, and file formats. One interesting challenge Kuny identified is establishing

digital preservation as a public good. In summary: we depend on historical records from the past to inform the present. Thus, it is necessary to continue to preserve our records for this purpose. The problem with preservation is that preservation itself only provides a benefit when the information is used. Successful preservation, then, requires making the data available and easily accessible.

2.1 Challenges in online digital archives

Providing access to the data involves its own set of challenges. Historically, material lived at a library and scholars journeyed *to* the library for access. Once there, they would interact with archival specialists to determine exactly which material they needed. Now, users expect access from any Internet enabled device. In fact—and perhaps largely in response to the efficacy of the Google search engine—we expect instantaneous answers to our poorly phrased information requests. The issue with such an approach in these digital collections is that users are frequently interested in discovery instead of search. In other words, they want systems to help them find something new and interesting to them, rather than locate something they already know. Consider a conversation that might have occurred at a library:

Librarian	How may I help you today?
Visitor	I've just moved here from Sweden. Is there a good book on local history?
Librarian	Oh, a lot of our early immigrants came from Sweden. I know just the book for you.

Our hope is to enable such assistance in a digital archive. Let us envision how this scenario might play out online to understand the challenges in providing access to digital archives.

User	Enter a query on "local history".
System	Provide a ranked set of responses to indicate the best references for information on local history; along with a list of major sub-topics including Swedish immigrants.
User	Click on the Swedish immigrants sub-topic list.
System	Provide a new set of ranked responses, with one highlighted as a "featured selection."

Consider the technologies necessary to enable this interaction. First, such a system must know that the query "local history" refers to the history of the area where the searcher is located or implies a particular local history. Second, the engine must have a means of searching on the topic or keywords associated with each item in the collection. Third, it needs a procedure to rank the results to provide a useful ordered list back to the user. Fourth, it must identify a set of sub-topics within the query.

For books, this situation is pretty well handled by existing tools and many libraries have been revising their *online public access catalogs* or OPACs to enable such searches. See the North Carolina State University Library website, the Queens Library website, and the Stanford Library website for examples:

http://www.lib.ncsu.edu/summon/
http://www.queenslibrary.org/
http://searchworks.stanford.edu/

The topic information in a book is often provided by the Library of Congress Subject Heading (LCSH) descriptors. For books published in the United States, the LCSH descriptors can be found in the first few pages of many books with the Library of Congress catalog data. For example, Nick Higham's book "Handbook of Writing for the Mathematical Sciences" [17] has the subject headings:

Mathematics–Authorship and Technical writing

indicating that the book deals with the issues of authoring mathematics and technical writing. These descriptors were an early type of indexing applied to books to enable card-catalogs to support subject lookup. Space in a card-catalog was limited, and thus the indexing needed to support a wide range of topics with an economy of index phrases. A more recent alternative is full-text search of the book, enabled by the increasing availability of born-digital content and large scale book scanning efforts. Together, these technologies support such searches for books, but leave room for future improvements. For example, the "local history" search we describe above is particularly problematic because "local history" is a particular type of history described in the Library of Congress subject headings. Such a search on these systems tends to return books about the concept of local history, one search result was a book about how to find out more about the history of your area, instead of books on the history of the area itself.

Digital preservation, however, goes far beyond books or digitized books. It encompasses both monumental and mundane digital artifacts. For these objects, subject heading data is unlikely to be available, and the items themselves may not be text. In particular, the Library of Congress, has over 14 million images. (Determined from the Library's web-page: http://www.loc.gov/rr/print/ access on 13 August 2010.) Other possibilities include: survey results, maps, audio, and video. The lack of textual description for these types of materials will feature prominently in the sections that follow because it is not always clear how we can best enable users to discover interesting artifacts. Our current techniques focus on extracting information from what little text we may have about the item.

2.2 Digital archives for historic material

Thus far, we've motivated the problem of accessing digital archives from the perspective of digital preservation. However, libraries are also a repository for many rare, culturally significant manuscripts, pictures, and other objects. These items are often fragile and not suitable to be widely handled; and yet the mission of a library is to share these items. Digitization and imaging provide an effective surrogate that may be widely shared. However, the same difficulties arise with accessing these items as with generic digital archives. Let us provide an example. During a visit to the manuscript division of the Library of Congress, one of their subject matter experts directed us to a box of John von Neumann's artifacts. Among these was a copy of his immigration card; see Fig. 1. Just like the goal of digital preservation is to find interesting material in a broad and diverse archive, the goal in these special digital collections is to

Fig. 1 A photo of John von Neumann's (Johann von Neumann) immigration card taken at the Library of Congress in January 2007; the original immigration card contains a photo from 1934. We consider this as an artifact that most mathematicians would be interested in *discovering* but would not know to search for

find the gems like—for us—John von Neumann's information. We would never have known to search for that ourselves. The key to finding these objects to is to have an thorough and rich environment of linked data.

2.3 Crowd-sourcing linked data

One fact that emerges from recent studies is that *simple* algorithms may perform as well or better than complicated algorithms when given additional data. See ref. [36] for an instance of this phenomenon in the Netflix recommendation problem (see the aside below) and ref. [14] for a thoughtful perspective on the role of data in computing. Here, we argue that *crowd-sourced* data is a suitable source for the rich context we need to enable a good virtual librarian system.

Netflix and the Netflix problem Netflix is a DVD-by-mail rental service in the United States and Canada. They have also expanded to offer Internet based streaming video access. A key challenge for them is recommending new DVDs and movies to their users. To motivate research in this area, they released 100 million anonymized ratings by their users, and offered $1,000,000USD to the first group to demonstrate a 10% improvement in predicting user ratings based solely on the existing set of user ratings.

The encyclopedia Wikipedia is perhaps the best example of open and crowd-sourced data. An open dataset is simply a dataset provided on the Internet without cost. One example of an open dataset is the website http://id.loc.gov/authorities, which provides an interactive exploration of the Library of Congress subject headings along with the ability to download the subject headings in bulk. The records behind LCSH, however, are still curated by the Library of Congress.

Wikipedia, in contrast, is an example of crowd-sourced data. Over the last 10 years, the encyclopedia was written and edited by a diverse group of unregulated individuals. They evolved a self-regulating mechanism that allowed almost anyone to contribute to the encyclopedia, while limiting the ability of individuals to manipulate its contents for their own purposes. Consider the difference from old models of information collection. Information repositories were supervised by a select group of experts, who would review and authorize changes in an attempt to avoid errors. In the case of LCSH, the process took decades and the rules for adding new things were

known only to a select few. As we shall see shortly, Wikipedia established a similar category system in just a few years (Sect. 4).

The success of crowd-sourcing is astonishing. It has become a pillar of so-called "Web 2.0" technologies. In a theory espoused by Surowiecki [40], the diverse perspectives of many people provide more reliable predictions than those of a few experts. This theory is known as the "wisdom of the crowds." A recent study on folksonomies, a common type of crowd sourced data used to describe items with a few short tags like on Flickr and Delicious (see the aside for more on Flickr and Delicious) shows that the tags produced by "verbose describers" are more useful than those from "categorizers" [27]. If we assume that experts are more likely to be categorizers, this could be taken as an empirical validation of the crowd-sourcing methodology.

flickr, delicious, and tagging The web sites flickr and delicious are photo sharing and bookmark sharing sites, respectively. They both feature a concept called *tags* that allowed users to annotate objects with short words or phrases. For example, a good tag for a picture of a flower is "flower" likewise. These tags are like the keywords on a mathematical paper, with the crucial difference that *anyone* can supply a tag, instead of just the author.

Regardless of the theoretical support, there is now a tremendous amount of data available from these more casual models of information collection. In Sects. 4 and 5 we explore using these data sources to generate new relationships between digital artifacts.

2.4 Definitions

To end our background, we present Table 1: a summary of acronyms and language used in this paper.

3 Understanding data and visualizing links

Digital archives already include many disparate data sources. None of these data share a common format. Our goal is to combine the data together to enable intelligent search and browsing, by using information from open sources to augment the incomplete metadata from a library record, for example. Table 2 presents an overview of the different data sets we use in this paper. There are three broad classes:

1. Library of Congress proprietary data,
2. open and crowd-sourced data,
3. multi-lingual data.

The first class contains information the Library of Congress does not typically share, such as the raw metadata behind the American Memory collection, or information the Library sells to attempt to recoup their cost. The second class of data is all freely available. This is the type of data described in Sect. 2.3. The final class is also Library of Congress proprietary, but has the distinct feature that the metadata is available in multiple languages. This paper focuses on the first two classes; although, we discuss ideas for the multi-lingual data in the section on future work.

Table 1 Services, Acronyms, and Definitions

Netflix	DVD-by-mail and video streaming service and website	Sect. 2.3
Flickr	photo sharing website featuring user-generated tags	Sect. 2.3
Delicious	bookmark sharing website featuring user-generated tags	Sect. 2.3
Wikipedia	a crowd-sourced encyclopedia	Sect. 2.3
Twitter	a micro-blogging message system with 140 character messages	Sect. 8.2
OPAC	Online Public Access Catalog	Sect. 2
MARC	MAchine Readable Cataloging	Sect. 3
XML	eXtensible Markup Language	Sect. 3
RDF	Resource Description Framework	Sect. 3
LCSH	Library of Congress Subject Headings	Sects. 3–4
HIT	a Human Intelligence Task	Sect. 7
born digital	content that never existed in anything besides a digitized form	Sect. 2
artifact	another name for the objects of a digital archive	Sect. 2
metadata	any information *about* a digital object, especially *time*, *place*, and *subject*	Sect. 1
Crowd-sourced	a term used to describe data collected from many unofficial sources	Sect. 4
Folksonomy	a specific type of crowd-sourced data consisting of a set of tags—short descriptions—applied to a set of objects in a database	Sect. 4
Tags	the lowest level of a folksonomy	Sect. 4

Each of these databases or collections has its own way of storing information, and there is even diversity within a collection. American Memory is actually a collection of collections. Some of the metadata associated with the items is in the MARC format; some of the metadata is in the XML format. We provide a sample from some of the raw information in these databases in Fig. 2. The details of the MARC [43], RDF, and XML formats are not relevant. Each data format roughly provides a set of records and fields about those records. Finally, some of the items may have annotations in yet another format. For example, the `mal` collection has metadata stored in XML files and annotations stored in SGML files (an XML predecessor). We mention all of these details and data formats to emphasize the heterogeneity of the raw data even at the lowest level. We must continually write new interpreters for each of these data collections to simply access the data itself.

Once we access the data, the problems multiply. In an ideal world each item would have a fully specified set of metadata including date, location, subject, and people specified in a consistent manner. Reality leaves much to be desired. We'll see how inconsistent some of the metadata inside these files are in Sect. 6. The other problem we encounter once we are able to read the data files is that we need to understand their contents. By understand, we mean to be familiar with the idiosyncrasies of a dataset—ideally just like an expert who has worked with the data for years. In the next section, we delve into the Library of Congress subject headings to illustrate one approach to understanding the contents of these databases.

```
<record>
    <leader>00760cam  2200253   4500</leader>
    <controlfield tag="005">20030904182120...
    <datafield tag="100" ind1="1" ind2=" ">
    <subfield code="a">Ladner, Joyce A.</su...
    </datafield>
    <datafield tag="245" ind1="1" ind2="0">
    <subfield code="a">Tomorrow's tomorrow:...
    <subfield code="b">the Black woman</sub...
    <subfield code="c">[by] Joyce A. Ladner...
    </datafield>
    <datafield tag="650" ind1=" " ind2="0">
    <subfield code="a">African American wom...
    </datafield>
    <datafield tag="650" ind1=" " ind2="0">
    <subfield code="a">African American fam...
    </datafield>
</record>
```

(a) An example of a MARC record in XML.

```
010    _amp 73117800
050 00 _aFLA 1783 (ref print)
050 00 _aFRA 4418 (dupe neg)
050 00 _aFRA 4419 (arch pos)
245 00 _aSt. Patrick's Day parade, Lo...
257    _aU.S.
260    _aUnited States :
       _bThomas A. Edison, Inc.,
       _c[ca. 1905].
500    _aCopyright: no reg.
500    _aDuration: 3:21 at 16 fps.
520    _aShows policemen and men in top
         hats and formal riding attire
         carrying large bunches of [...]
650  0 _aParades.
650  0 _aSaint Patrick's Day.
650  0 _aHolidays.
651  0 _aLowell (Mass.)
710 2  _aThomas A. Edison, Inc.
```

(b) The MARC metadata for a record in the motion picture collection (*papr*) of American Memory translated into text.

```
<div id="d0004200">
<p><hi rend="underscore">
From Leander Munsell to Abraham Lincoln
April 23, 1846</hi></p>
<p>Paris  Apr  23 /46</p>
<p>Dear Sir</p>
<p>I trouble you a with a line in regard
to the Grandview Office&mdash;  From my
knowledge of the condition of things at
Grandview I have no doubt that J. V. Brown
should receive the appointment of P. M.
at that place before Mr Payne or any
other known applicant&mdash;  If you will
aid his appointment you will oblige me
&mdash;</p>
[...]
<p>Sincerely Yrs</p>
<p><hi rend="underscore">L. Munsell</...
</div>
```

(c) A extract from the SGML annotations in the Abraham Lincoln papers (*mal* collection).

Fig. 2 Three examples of our data files. This peek into the guts of each file shows the raw form of the library records

Table 2 A summary of the data used in our explorations. For each collection, we list its size in terms of the number of "things" inside the collection. Note that American Memory is a group of collections. Thus, `papr`, `mal`, `gmd`, and `wpa` are sub-collections inside American Memory. The mixed formats in American Memory are MARC and XML; the mixed formats in Global Gateways are structured text and MARC

Type	Collection	Total Recs.	Format	Notes
Library of Congress Proprietary	Subject Headings	298,964	MARC	Authority files from Dec. 2006
	Name Authorities	6,662,688	MARC	Authority files from Dec. 2006
	Catalog	7,207,747	MARC	Library of Congress Book catalog
	American Memory	617,673	Mixed	101 Heterogeneous Collections
	— `papr`	703	MARC	Motion pictures
	— `mal`	20,158	XML	Abraham Lincoln papers
	— `gmd`	6,888	MARC	Maps collection
	— `wpa`	2,000	XML	American life histories
Open and Crowd-Sourced	Wikipedia	3,799,337	XML	(From April 2007)
	Wikipedia Categories	226,221	(derived)	(From April 2007)
	Geonames	6,914,549	Text	A gazetteer
	Project Gutenberg	24	Text	Full text books
Multi-lingual	Global Gateways	21,274	Mixed	
	World Digital Library	196	XML	

3.1 A graph visualization of LCSH: the subject heading galaxy

The Library of Congress Subject Headings (LCSH) is a database of terms, maintained by the Library of Congress, for use in indexing the subject matter of bibliographic records and also in cross-referencing between related subjects. A subject heading has broader terms, narrower terms, and "see also" terms. For example, the subject heading "Mathematics" is related to the broader term "Science" and the narrower terms "Algebra", "Economics, Mathematical", and "Women in mathematics". We can view the LCSH database as an undirected graph where each subject heading is a vertex and each relationship defines an undirected edge.

To help quickly build our understanding of the information contained in these links, we wanted to visualize the graph. Two common ways to visualize a large graph are (i) to visualize small regions of the graph [35]; or (ii) to visualize the entire graph. We worked with both techniques and only describe the second in the interest of space. The insight we gain by visualizing the entire graph is a sense of the overall linking structure of the network, from which we may be able to pose more pointed questions. For an example of this type of analysis, see ref. [20] for insights into the Twitter network. To visualize the graph, we need a means of computing a layout—an assignment of points to coordinates in the plane—of a graph with hundreds of thousands of nodes. This is a challenging computation and an active area of research—see ref. [22] for a recent contribution in large graph visualization.

We used the Large Graph Layout (LGL) program [1], which proceeds roughly as follows:

1. find a minimum spanning tree for the graph, and use this representation of the graph until step 3;
2. find the vertex with minimum total shortest-path distance to all other vertices—we call this vertex the center;
3. for $k = 1$ to $\ldots$, add all the vertices k edges away from the center vertex and locally optimize their positions based on the minimum spanning tree;
4. do a final refinement based on the edges of the original graph.

We choose LGL because of step 1. Based on preliminary work with the graph behind LCSH, we discovered it may have significant regions of tree-like structure. (This analysis involved the size of the 1-core of the graph [38], which is a measure of how many vertices are removed when iteratively removing vertices of degree 1.) For this reason, a layout algorithm that exploits tree-like structure in the graph should produce useful structure.

In the LGL process, most of the work is in the final step. It involves simulating a set of dynamics to compute an approximate minimum energy state. Please see the LGL paper for more detail on step 3; we will focus on step 2. When we started working with the code, the step of finding the center took around two hours. It solved a breadth-first search problem for each vertex in the graph. This implementation did not utilize the structure of the tree in computing the center vertex. Many graph algorithms greatly simplify in the presence of a tree structure. The same simplification occurs for this problem as well, and the sum of all shortest paths satisfy a straightforward recurrence in a tree. The implementation of this recurrence requires only requires work equivalent to three breadth-first searches. After implemented this technique, step 2 took only seconds to compute.

To describe our optimization to efficiently compute the center, let $D_{u,v}$ be the number of edges in the shortest path between vertices u and v in the minimum spanning tree. We want to find the vertex c that minimizes $\sum_v D_{c,v}$. The key idea to our optimization is that there is a unique path between any two vertices in a tree. Formally, the procedure is as follows. Call $C_u = \sum_v D_{u,v}$ the center score of vertex u. In a tree, consider changing C_u to C_w where we have an edge (u, w). All we need to do is count how many paths starting at u get longer when they start at w instead, as well as how many paths get shorter when they start at w. See Fig. 3 for an example of how we would take advantage of this observation. To implement this observation for a tree T, pick an arbitrary vertex a, and "root" the tree at vertex a. Next compute C_a. For every vertex w connected to vertex a, we have

$$C_w = C_a - N_w + (n - N_w)$$

where N_w is the number of vertices in the subtree rooted at w and n is the total of vertices. To explain this formula, consider that all paths from w to vertices in the subtree starting at w are one edge shorter, and so we reduce C_a by N_w. Additionally, all the other vertices in the graph, $(n - N_w)$ in total, have paths that are one edge longer when started at w instead of a. By repeating this procedure for all subsequent levels, we can compute C_v for each vertex v in linear time. The entire process requires three passes over the graph: the first to compute C_a for the arbitrary root; the second to compute N_v, the size of each subtree for each v; and the third to compute C_v given C_a and N_v for each vertex.

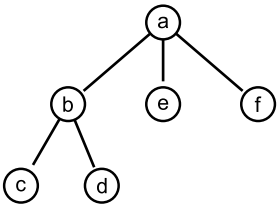
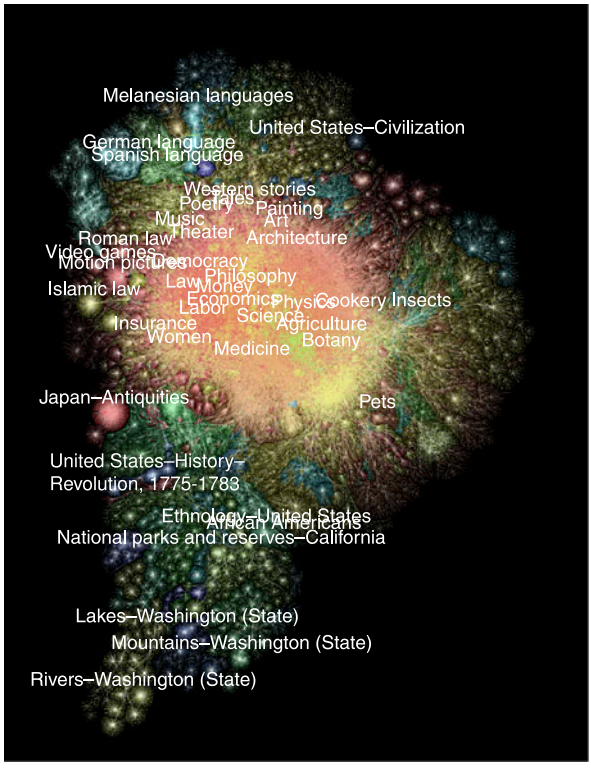


Fig. 3 A small example of how to compute the total shortest-path distance to all vertices efficiently in a tree. We can easily compute the total shortest-path distance starting from the arbitrarily chosen root vertex a: $C_a = 7$. Now to compute C_b, we note that there are three vertices that become one edge further away—a, e, f and three vertices that become one edge closer—b, c, d. Thus, $C_b = C_a - 3 + 3 = 7$. Likewise, we find $C_c = C_b - 1 + 5 = 11$, and the same for C_d. Both C_e and C_f are also 11. In this example, either a or b can be the root vertex. Also, $N_a = 7$, $N_b = 3$, and $N_c = N_d = N_e = N_f = 1$

Fig. 4 A visualization of the graph behind the Library of Congress subject headings. This drawing shows the largest connected component of the undirected graph of links in the Library of Congress subject headings, with nodes colored based on a clustering from the CLUTO program, and a few node labels shown to illustrate the topics in a particular region



After making these changes, we ran the LGL algorithm on the largest connected component of the undirected graph of LCSH. We present a visualization using the layout computed by LGL in Fig. 4. Edges are drawn with alpha-blending to show the local density. Each node is colored based on a clustering computed using the CLUTO program [23]. Note that we see large regions with the same color. This means that

both CLUTO and LGL are identifying similar structures in the graph. For a more in-depth look at this visualization, see

http://cads.stanford.edu/lcshgalaxy

Based on this visualization, we find the following structure in the LCSH network. There is a dense core of general interest subject headings such as "Law", "Science", and "Art". Around this core we find a set of more esoteric topics, including an extensive region of geographic features, which forms the southward extent from the yellow core. Another insight is that some regions are perhaps better categorized than others. At the left hand side of the figure is a large star-like construction centered around the subject heading "Japan–Antiquities". There are over one thousand subject headings in this star, with only a single connection back to the star's center. In contrast, other regions of the graph (such as the language subject headings in the upper left) show better organizations.

During our explorations of this visualization, we noticed a few properties about the graph that reminded us of another graph: the category structure of Wikipedia. In the next section, we elaborate on this relationship.

4 Utilizing open crowd-sourced data such as Wikipedia

Recall the structure of the Library of Congress subject headings from the previous section. Each subject heading is related to others by "Broader term", "Narrower term", and "See also" references. We interpret these relationships as an undirected graph. The category pages in Wikipedia have a similar structure. Every page in Wikipedia is a member of one or more categories. For example, the page about "Singular Value Decomposition" belongs to the categories "Linear algebra", "Matrix theory," and "Functional analysis". Categories have sub-categories and related categories, which form a hierarchical structure with a few additional edges. It may seem surprising, but the undirected graph of Wikipedia categories has a similar number of vertices to the graph of LCSH—205,948 vs. 297,266. Other properties are similar too: the largest connected component size is about 150,000 vertices in both graphs, the average distance between any pair of nodes is around 7 in both, and around 6,000 nodes have *identical* textual labels.

Based on these results, we wanted to *match* or *map* each vertex in the LCSH graph to a vertex in the Wikipedia graph. The idea behind finding a match is that the experts developing the LCSH can use the matches to improve the coverage in new or rapidly evolving areas that, potentially, have better coverage in Wikipedia. See Fig. 5 for an example. We formalize the problem as a sparse network alignment problem [2], whose solution tells us how to match the vertices of two graphs when we have a reasonable set of *potential matches* between them. In Fig. 6 we show the structure of a network alignment problem. Also in ref. [2], we propose a message passing algorithm for this case. Our algorithm produces nearly optimal solutions to the network alignment problem with LCSH and Wikipedia in a few minutes (even when implemented in Matlab) [2]. In Table 3 we list a few matches identified by this approach. The matches are organized in three groups, correct, mildly incorrect, and

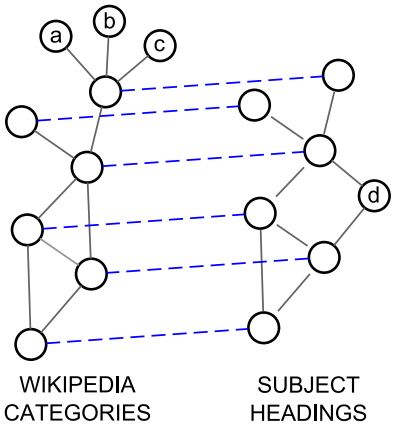


Fig. 5 We hope to match the categories of Wikipedia to the subject headings from the Library of Congress. Using this 1-1 matching (the *dashed* links), we can suggest new subject headings (the nodes a, b, and c) and add information to Wikipedia categories (the node d should tell us something useful about its matched neighbors)

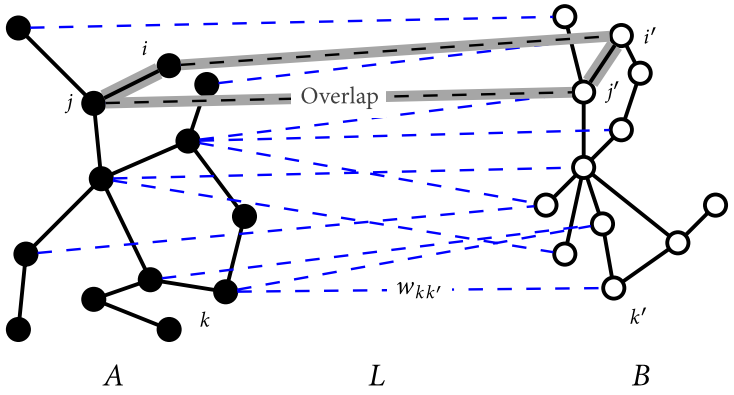


Fig. 6 In the general network alignment problem, the goal is to match the vertices of graph A to the vertices of graph B, and trying to *overlap* as many edges as possible and maximize the weight of the edges in the matching $\sum w_{kk'}$. Formally, we get an overlapped edge for a matching when (i, j) is an edge in A and its image under the matching, $(m(i), m(j))$, is also an edge in B

Table 3 Results of matching of LCSH to Wikipedia. See the discussion in the text

LCSH ↔ Wikipedia
Correct
Dollar, American (Coin) ↔ United States dollar coins
Web sites ↔ Websites
Environmentalists ↔ Environmentalists by nationality
Peninsulas–Southeast Asia ↔ Peninsulas of Asia
Mildly incorrect
Cosby family ↔ Bill Cosby Songs
Peasants in literature ↔ Peasant foods
Incorrect
Hot tubs ↔ Hot dogs
Masques ↔ Vampire: The Masquerade

incorrect. The mildly incorrect set of matches are ones that are "nearby" and refer to related, but different, concepts. In our current formulation of the problem, there is no penalty for placing a match that does not accomplish much—thus our results are littered with many spurious matches. We hope to incorporate a penalty term in future work.

While we designed our algorithm to work on problems with hundreds of thousands of nodes, there are other successful techniques to match vertices in graph. This problem occurs in pattern recognition, see ref. [5] for a survey of that work. There are many matrix problems that arise as well. See refs. [3, 11, 12, 33, 34] for examples.

More generally, the problem of combining linked data is known as ontology matching or ontology alignment. An ontology is a set of statements that express relationships in a structured form. They are often described as a set of statements with a subject, verb, and object. Consider the Wikipedia categories previously mentioned in this section. As an ontology, they would be expressed:

subject	verb	object
Singular Value Decomposition	*is in category*	Linear algebra
Singular Value Decomposition	*is in category*	Matrix theory
Linear algebra	*is related to*	Affine geometry
Linear algebra	*is a subcategory of*	Algebra

Algorithms for ontology alignment often include greedy approaches to optimize a similar objective to our network alignment approach [9, 18].

5 Resolving ambiguous references

The temporal and geographic context of an item are essential metadata for discovering interesting related information. Both types of information provide an easy way to browse a collection or to relate two different artifacts. However, not all the items in American Memory had reliable *place* metadata, and one of the problems we faced was extracting the geographic entities from a book or manuscript. The more general problem of deriving structured information from unstructured sources is known as information extraction [6, 32]. What we explain in this section is a special case of geographic information extraction. We develop a technique that can easily be used for temporal and event references as well. Moreover, in the conclusion of the paper, we propose an extension of our algorithm for the general problem of named entity disambiguation.

In our approach for extracting and disambiguating place names, we assume there is descriptive text available, or some means of getting descriptive text—perhaps via speech recognition, optical character recognition, or crowd-sourced tags. The first step is to extract a list of location names from the text. A location name is a specific type of a named entity. Given a collection of text, a *named entity recognizer*, or NER, can be customized to only output the strings of text likely to be the name of a location. We used the freely available Stanford NER [10]. To do the disambiguation, another piece of information is required: the actual geographic coordinates of a location name. A database of mappings between geographic coordinates and location

names is known as a gazetteer. We used geonames as our gazetteer. Geonames is a freely available collection of around 7 million place names and the latitude and longitude of each location. Together, we have a collection of place names from the Stanford NER software and a collection of coordinates from geonames. We have almost accomplished our goal of finding all the places mentioned in a book or manuscript. However, place names do not uniquely map to locations. Context, in the form of other place names nearby, usually provides a solution.

Let us pose a mathematical formulation of the problem using context. Let $X = (x_1, \ldots, x_n)$ be a sequence of locations mentioned, ordered by their position in the text—this sequence is the output of the NER software. Formally, the location name x_i preceded x_j if $i < j$. For each location x_i, we assume that there is a set $Y_i = (y_{i,1}, \ldots, y_{i,k})$ of known locations that match the textual reference x_i. These sets Y_i correspond to all matches in the geonames database for a location name x_i. We call the set of all possible candidates $\mathcal{C}$, and so each $y_{i,r} \in \mathcal{C}$. We further assume that we have a distance function between elements in $\mathcal{C}$. At the moment, think of D as the geodesic distance between the latitude and longitude of each location. For other types of problems, D should change; for example, it might be the temporal difference between two events for resolving ambiguous event names. Nonetheless, let $D : \mathcal{C} \mapsto \mathbb{R}$ be this function. Our goal is to choose a *single* reference for each candidate. One natural way to pick these references is to minimize the distance between the locations mentioned. This idea translates into the optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n-1} D(z_i, z_{i+1}) \\ \text{subject to} \quad & z_i \in Y_i \quad \text{for all } i. \end{aligned}$$

In this formulation, the disambiguated locations are $(z_1, \ldots, z_n)$. To solve this problem, we can use a dynamic program. Let $f_{j,r}$ be the optimal solution of

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{j-1} D(z_i, z_{i+1}) \\ \text{subject to} \quad & z_i \in Y_i \quad \text{for all } i \\ & z_j = y_{j,r}. \end{aligned}$$

Then

$$\min_{s \in Y_j} \big(f_{j,s} + D(y_{j,s}, y_{j+1,r}) \big).$$

Clearly, $\min_{r \in Y_n} f_{n,r}$ is the minimizer to the original problem. This greedy algorithm requires $d = \max_j |Y_j|$ work for each computation of $f_{j+1,r}$. There are at most d such computations for each j, and thus the total work of the algorithm is bounded above by nd^2. In practice, d should be fairly small as most geographic entities will have nearly unique identifiers.

One concern with this algorithm is that it could easily be fooled into making the wrong decision by a single distant reference. Consider the following passage:

> A British holidaymaker was sent to San Juan in Puerto Rico, rather than San Jose in Costa Rica by her travel agent, and other tourists aiming for San Jose, Costa Rica have landed in San Jose, California, and have then had to ask the way to San Jose.[2]

The algorithm above will assert that the final reference to "San Jose" refers to "San Jose, California" because that has distance 0, which is incorrect. A straightforward fix is to incorporate additional pairwise comparisons. Consider the generalized problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{\substack{0<j-i\le T \\ 0\le i,j\le n}} D(z_i, z_j) \\ \text{subject to} \quad & z_i \in Y_i \quad \text{for all } i. \end{aligned}$$

Again, we can solve this problem using a variation on the previous dynamic program. We show the generalization for $T = 2$ and note that larger context sizes are easy to derive. Let $f_{k,(r,s)}$ be the optimal solution of

$$\begin{aligned} \text{minimize} \quad & \sum_{\substack{0<j-i\le 2 \\ j\le k}} D(z_i, z_j) \\ \text{subject to} \quad & z_i \in Y_i \quad \text{for all } i \\ & z_{j-1} = y_{k-1,r} \\ & z_j = y_{k,s}. \end{aligned}$$

Then

$$\min_{w\in Y_{j-1}} \big(f_{k,(w,r)} + D(y_{k-1,w}, y_{j+1,s}) + D(y_{k,r}, y_{j+1,s})\big).$$

Now $\min_{(r,s)\in Y_{n-1}\times Y_n} f_{n,(r,s)}$ is the minimizer to the $T = 2$ problem. Let us return to the "San Jose" example above to show how this helps. There are five geographic references: "San Juan in Puerto Rico", "San Jose in Costa Rica", "San Jose, Costa Rica", "San Jose, California", and "San Jose." Only the final reference is ambiguous, suppose we only consider San Jose, California and San Jose, Costa Rica as possible alternatives. As we vary T, consider the outcomes:

$T = 1$	San Jose, California
$T = 2$	San Jose, California *or* San Jose, Costa Rica
$T = 3$	San Jose, Costa Rica
$T = 4$	San Jose, Costa Rica.

Thus, using moderate T makes the algorithm less sensitive to outliers.

The algorithm often yields satisfying results, yet it has some weaknesses. First, the assumption underlying the optimization problem is that the geographical references in the text tend to form small clusters. Furthermore, it assumes that consecutive locations should be geographically close. These assumptions may not always hold.

[2] Accessed from http://www.skyscanner.net/news/articles/2010/09/007959-destination-doppelgangers-same-name-different-country.html on 8 September 2010.

Second, geodesic distance is only a proxy for the probability that two locations are mentioned nearby. Consider the sentence: "I just flew from New York to London." It's almost surely the case that the author flew from New York City, New York to London, England, and not from New York City, New York to London, Ohio, or from New York, Lincolnshire to London, England, both of which are geographically closer. To solve this issue, we need an improved distance function between locations. We also may need to include additional context into the algorithm. Please see the conclusion for a potential improvement to this algorithm.

6 Metadata and title remediation

As we mentioned in the introduction, we do not have full text for many of the items we wish to work with. Another approach is to try and extract information from the metadata itself. There are often oblique references to place names or dates in the metadata, and we could use those as a surrogate instead. Using metadata to enrich itself is known as metadata remediation [7]. We first discuss remediating the date field of a metadata collection. The date field is particularly important because people often wish to browse for items based on their temporal relevance.

6.1 Simple remediation

The idea of remediating metadata with itself may seem strange. After all, the point of metadata is to provide structured information about an artifact. How can we possibly improve it? We can indeed do so because the metadata may have been entered inconsistently. Let us show an example. For the collection *gmd* in American Memory we examined all the elements of the MARC field that should contain the date information, e.g., 260$c (date of publication). A summary is shown in Table 4. These entries—as is—are wildly inconsistent and unsuitable for use to display a list of items relevant to a particular year or range of years. To correct these entries, we adopted an ad-hoc solution. In each of the patterns we found, the year information is almost always indicated by the #### string. Thus, to standardize the metadata, we converted these years into a standard date format and output the corrected metadata.

6.2 Title remediation

Another challenge the Library faces with many of these collections is that the metadata must be refined over time. During the initial digitization of the *papr* collection of early motion pictures, the digitizers only collected a wordy summary of each video. See Fig. 7 for an example. Most modern online displays, such as YouTube, often require a short title for each item. These titles must be snappy and searchable to attract interest. Unfortunately, the existing descriptions were too long to serve as titles. Because there were less than one thousand videos in this collection, the Library manually shortened each description into a title. We asked: can we automatically shorten the descriptions and extract a good title? Again, see the figure for an example of our title on that same video compared with the Library's title. The generated title

Table 4 The table at left lists a format of a type of date pattern and an example of that pattern. The patterns are shown in four groups: obvious, ambiguous, other calendars, and wrong. Bunka 1 refers to the first year of the Bunka era in Japan, which is the year 1804. Likewise, Guangxu 30 is the 30th year of the Guangxu era in China, that is 1904. We abbreviated between as btw. for brevity in the table

Format	Example
####-##	1601-15
####-####	1862-1863
[Month] #, ####	Decr. 1, 1793
btw. #### and ####	btw. 1755 and 1762
#### [Season]	1939 Spring
anno ####	anno 1668
##/##/##	03/02/64
###-?	184-?
Bunka # ie ####	Bunka 1 ie 1804
Guangxu ## ####	Guangxu 30 1904
#########	185000930
United States	United States

succinctly captures the major essence of the video. We discuss how we evaluated our generated titles in the next section because this raises a few other points we wish to highlight.

Let us begin by summarizing the process of generating titles. The following sections cover these points in more detail. Let $\mathcal{C}$ be a large, background collection of text. This collection helps to find discriminating phrases for a title, which is discussed in Sect. 6.4. Next, let $\mathcal{T}$ be a collection of sample titles. For each title $t \in \mathcal{T}$, compute the part of speech sequence for the title using the Stanford part of speech (POS) tool (or any part of speech identification tool). Construct a set of title templates from these part of speech sequences as described in Sect. 6.3. Then, given a description, compute the part of speech sequence for this description. For each bigram (two-word sequence) in the description, compute the phraseness and informativeness scores for it as in Sect. 6.4. Take the sum of these scores as the overall keyphrase score for this bigram. For each title template, construct a sequence of bigrams that matches the part of speech sequence. Pick the title with the highest sum of scores of its constituent bigrams. We summarize the process in Fig. 8.

6.3 Title templates

The process begins by identifying common part of speech patterns in an existing database of titles. These patterns are forms like

Excavating	for	a	New	York	foundation
VBG	IN	DT	NNP	NNP	NN

where the codes stand for: verb in gerund form, proposition, determiner, proper noun, proper noun and noun, respectively. We computed these using the Stanford part-of-speech tagger [42]. The idea is that a large collection of titles will have common patterns in the part-of-speech sequences. We can identify the most common patterns

Summary Shows policemen and men in top hats and formal riding attire carrying large bunches of flowers as they parade on horseback. Angle changes slightly as a marching band carrying a drum inscribed Bugle Corps, Lowell appears, followed by a uniformed military group carrying rifles and marching in formation. Angle changes to show carriages and the rest of the procession. Scene changes to a building, pans to the steps, and shows a clerical figure in robes exiting from the church and waving his hat in acknowledgement. No titles.
Manual title St. Patrick's Day parade, Lowell, Mass.
Our title men parade on horseback

Fig. 7 An example of our automatic title generation on a video taken by Thomas Edison in 1905 of a parade. The video is now available from YouTube: http://www.youtube.com/watch?v=mKzcjKDgxHY

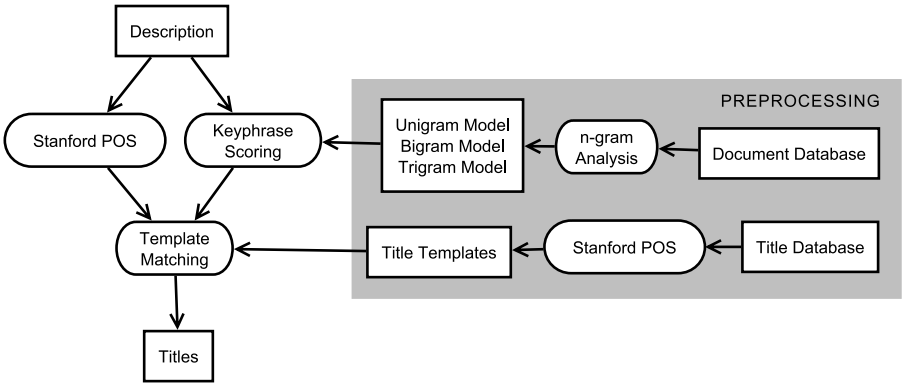


Fig. 8 Our title construction process. *Circles* indicate processing and rectangles indicate data. The *gray area* indicates one-time preprocessing. The remainder of the process must be done for each description

and use them as title templates. We can then match text from the description to the title templates and hope the results are useful titles. Thus, the first step in our title generation routine is to compute a set of title templates. We used the newswire collection for this task. This collection contains 1.3 million articles. For the title of each article, we computed the part of speech sequence and analyzed the patterns. The result is a database of 225,000 title templates.

6.4 Phrase scoring

To build meaningful titles, we need to extract meaningful phrases from the description. We use an idea from ref. [41]. Their process to find them involves scoring a sequence of words based on two measures: the informativeness and the phraseness. A sequence has high informativeness if it is very unlikely to occur in normal text. An example would be a description that the "singular value decomposition." This sequence of words is exceedingly unlikely to occur in day-to-day text, and thus this phrase is highly informative. However, note that "singular value decomposition" is likely to appear in papers in the SIAM Journal of Matrix Analysis. Thus the informativeness of this phrase is relative to a background collection of "standard text." A sequence has high phraseness if the statistical properties of the sequence radically change when we separate the phrase. The phrase "New York" has high phraseness because a document about "New York" will almost always mention "New" and "York" together, and thus, the statistics of "New" and "York" will be coupled in all documents.

These concepts are formalized by measuring the empirical probability distribution of sequences of one, two, and three words in a background selection of text. Let $\mathcal{C}$ be this background collection. The choice of $\mathcal{C}$ will impact what words are chosen as important as in the "singular value decomposition" example above, but not what are considered phrase-like. Each $d \in \mathcal{C}$ is really a sequence of word tokens $d = (w_1, \ldots, w_m)$. A unigram distribution is the probability of each single word in the collection of documents. A bigram distribution is the probability of each sequential pair of words in the collection of documents. A trigram distribution is defined in the same way. These unigram, bigram, and trigram distributions are sometimes called *language models*.

Now, consider the sequence of words in the description of an item: $d = (w_1, \ldots, w_m)$. For a sequence of words (w_i, w_{i+1}, w_{i+2}), the phraseness score is

$$P(w_i, w_{i+1}) = \text{Prob}\big[(w_i, w_{i+1}, w_{i+2}) \text{ in } d\big] \cdot \log\left(\frac{\text{Prob}[(w_i, w_{i+1}, w_{i+2}) \text{ in } d]}{\text{Prob}[(w_i, w_{i+1}) \text{ in } d] \cdot \text{Prob}[(w_{i+1}, w_{i+2}) \text{ in } d]}\right).$$

The informativeness score is

$$I(w_i, w_{i+1}) = \text{Prob}\big[(w_i, w_{i+1}) \text{ in } \mathcal{C}\big] \cdot \log\left(\frac{\text{Prob}[(w_i, w_{i+1}) \text{ in } \mathcal{C}]}{\text{Prob}[w_i \text{ in } \mathcal{C}] \cdot \text{Prob}[w_{i+1} \text{ in } \mathcal{C}]}\right).$$

These scores are just the Kullback-Leibler divergence measures between the trigram probability and bigram probability in the description for phraseness and between the

bigram probability and unigram probability in the background collection for informativeness. With extremely short descriptions, we use the background collection to compute phraseness scores instead of the description text itself.

One problem with these empirical probability distributions is that we may encounter word sequences that did not appear in the background collection. These novel events should not have zero probability in the above formulas, and smoothed probabilities are a standard correction for these seemingly zero probabilities. A simple type of smoothing used in statistics is known as a pseudo-count, and the classic example is Laplacian smoothing, based on Laplace's rule of succession. For the probability of n-gram occurrences in language, two common techniques are Katz smoothing [24] and Kneser-Ney smoothing [26]. In Katz smoothing, the measured counts are discounted by a multiplicative factor less than 1. The removed counts are distributed among the unobserved n-grams based on lower-order n-gram counts, for example, unigram counts instead of bigram counts. Kneser-Ney smoothing uses additive discounting instead of multiplicative. It also includes a better way of constructing lower-order n-gram models that handles multi-word combinations better. For example suppose "San Francisco" is common, but "Francisco" occurs only after "San" Kneser-Ney gives "Francisco" a lower unigram probability because it only appears in certain bigram combinations, which is captured in high bigram probabilities.

Putting it all together Once we have the scores on the utility of a particular phrase, all we need to do is match the phrases with the title templates to generate a title. The title with the highest weight (sum of scores) is likely to be the best title.

Now let's consider how we evaluate results in this paper, including these titles.

7 Quality assessment challenges

This paper presents numerous results: a visualization of the subject headings we claim is *useful*; *interesting* matches between the subject headings and categories; *correct* geographic references; *better* titles. Notice the emphasized words. A fundamental challenge in working with digital archives is that many problems do not have objective results. Subjective evaluation must suffice. Put another way, the ground truth is not the solution of a mathematically defined problem, but rather what someone asserts is true. To gather consensus on a good result when even the right answer might be unclear, we ask multiple people.

We will now discuss how to evaluate the set of titles we generated in the previous section. In brief, we outsourced the job to Amazon's Mechanical Turk. Amazon hosts a service named the Mechanical Turk in honor of the infamous faux-automaton chess player of the end of the 1700s.[3] Amazon's service allows people to post *human intelligence tasks* or HITS for short. A HIT is a small task along with a small reward. Finding the business name on a web-page is a HIT that could make someone $0.05USD. Users can post a HIT and pay a small fee, or accept a HIT and earn a small

[3] It seemed to be a machine but actually had a human hidden inside.

reward. The aggregate wage of a worker on Mechanical Turk could be a few dollars an hour. For some, it has even become a required source of income [37].

Given the economical nature of Amazon's Mechanical Turk system, research on using it for user studies has flourished. Ref. [25], investigated whether workers could produce ratings of Wikipedia articles that match those from Wikipedia editors—they can; and ref. [16] investigated whether graphical perception experiments among workers match known results from controlled laboratory experiments—they do. In terms of the workers, a recent study concluded that they were becoming "increasingly international" [37]. And while an obvious problem with Mechanical Turk results is that it is easy for a worker to game the system and provide fraudulent results, another problem is that workers suffer lower wages because there are few mechanisms to protect them [39].

In our study, we generated titles for 20 different descriptions using the algorithm in the previous section. In these cases, we used the best scoring keywords. Our HITS asked people to choose the *most informative* title between our title and the Library generated titles. The HIT provided the description text for reference. In total, we asked for 20 evaluations of each description and paid $0.02USD for each evaluation. The evaluation took less than one hour to complete after posting. Our automatically generated titles were selected in 80% (320 of 400) of the HITS. Note that we cannot reuse the data we collected from these workers because they are specific to a single comparison between titles. In the case of the geographic disambiguation, we could reuse the answers from the workers.

However, the two evaluations described above are simple in comparison with evaluating a search and browse system. Consider the differences. Above, when choosing titles, it's straightforward for workers to pick a preferred title. They may not have a clear preference, in which case the choice will be somewhat random, but there is no difficulty in specifying the answer. For the complete system, workers would need to identify resources they find interesting. Yet how are they to discover this information without evaluating the entire set of artifacts? One idea would be to ask workers if a few items are interesting given a hypothetical setup. But our goal is to identify items with subtle correlations to the user, as in the local history example from the background section. Connections of these types are difficult for others to evaluate. Thus, evaluating these complete systems is a challenging problem. We discuss an approach using user feedback in the conclusion.

8 Conclusions and ideas for future research

Recall our motivation. Modern collections of digital data require novel search technologies to make them relevant and worth storing. Historical collections of digitized data require sophisticated discovery methods to get people to the artifacts they find interesting. Both of these scenarios require interesting metadata about the objects of the digital collections. In this paper, we presented a global visualization of the Library of Congress subject headings (Sect. 3). This visualization helped us rapidly understand a new collection of linked data. Based on our experience with this dataset, we next explored an algorithm to *match* the subject headings in the Library of Congress collection with the categories in the Wikipedia encyclopedia (Sect. 4). Our algorithm,

described in ref. [2], produced near optimal theoretical results, as well as a potentially useful set of matches between the two datasets.

Next, we looked at the disambiguating geographic references in a text (Sect. 5). This led to a simple dynamic program using the distances between possible locations that we can easily solve. Geographic reference resolution is a particular case of metadata remediation, and we continued with another exploration of constructing better titles given short descriptions (Sect. 6). Finally, we described how to evaluate our approaches with Amazon's Mechanical Turk (Sect. 7).

These ideas only probe some of the possibilities. In the remainder of this section, we discuss two ideas we are currently exploring and three ways to forward research in the area of computational approaches to digital stewardship.

8.1 Extensions

Let us briefly mention two extensions of our work.

Recall the setup of the geographic disambiguation problem. We assume that the location names had known locations associated with them. What if, instead of only considering the set X of location references, we consider the full set X of all *named entities*? A named entity is person, place, or thing. We can still apply our disambiguation approach, but with a few changes. First, the notion of distance must now include places and things. To define such a distance, we restrict ourselves to the named entities in Wikipedia and use the graph distance in Wikipedia. While there a few possible choices of graph distance in Wikipedia, we use the number of edges in the shortest path between the nodes. An alternative is the commute time distance between nodes [13]. Using the same approach on this new data, we can disambiguate the people, places, and things mentioned in a book or manuscript. In this case, the use of Wikipedia is intentional. The link structure of Wikipedia is known to be correlated with semantic relationships [46]. One further improvement is to consider *all* connections between named entities. For any given T, the optimization problem we were solving only considered distances to nearby references. Using all connections makes the problem NP-hard. However, we can formulate it as a network alignment problem and use our scalable message passing solver [2]. The result is a tool to disambiguate named entities in text. Preliminary tests of the tool show that the output is more accurate.

Second, recall that our title generation procedure produced important keyphrases before the title matching. These keyphrases may themselves be useful for navigating through document collections. To get them, we simply stop the title generation procedure before the title matching. We applied this idea to the American Life Histories collection, and the initial results are promising. This collections consists of small written synopses of the lives of Americans between 1936–1940. Our extracted keywords serve as a means to navigate the thousands of histories by grouping histories with shared keywords. This is a form of faceted searching [45], which is a key browsing technology for information discovery.

8.2 Future directions

We'd like to end with a few broad research directions that we feel will be important to search and discovery in digital archives. Our hope is to inspire future work in this

area. We discuss three possibilities: multi-lingual search and discovery, explicit and implicit user feedback, and Twitter.

Multilingual Although English is widely viewed as an "international language", the native English speakers only account for around 5% of the world's population. Even the totality of English speakers is only 15%.[4] Suffice it to say that ignoring 85% of the world's population is not a viable stratagem for success in archiving all digital data. Many of the items are not natively expressed in English. In our data section, we mentioned two datasets with multi-lingual content: Global Gateway and the World Digital Library. Metadata in these datasets is in two or more languages. The challenge is identifying the best way to search and browse these collections without translating every item and its metadata into every possible language, which is the current strategy. A promising approach is to use a variant of latent semantic indexing [8] with a PARAFAC2 tensor factorization [15] and a multilingual parallel corpus [4]. A multilingual parallel corpus is a set of sentences or documents translated into each language. One popular choice of such a corpus is the Bible, which is available on a verse-by-verse basis in nearly every language. The output from this setup is a multilingual concept space that is searchable in any language from the multilingual corpus.

Feedback One aspect of a virtual librarian system we have not yet discussed is using user feedback. There are many forms that user feedback assumes. In crowd-sourced systems, users often contribute directly; in recommendation systems, users score the system's responses; and in search systems, users click links. These span a gamut between explicit (crowd-sourced and recommendation) to implicit (search). Determining the best way to utilize this feedback in a digital archive is not known. One of the constraints of digital archives is that some of the material is culturally significant. Thus, any feedback system must have stringent guards against malicious user behavior. No one wants to see erroneous connections between such material on national websites. An obvious use for either implicit or explicit feedback is evaluating the system. As we mentioned in the section on quality, evaluating a complete discovery system is a challenging endeavor. Using implicit feedback is actually one of the ways that Google continually improves their search engine [31]. In the case of a real-world library system, implicit feedback could track which users follow which of our suggested connections between items, or how long users spend looking at the information on a page. These are plausible surrogates for asking users if they have discovered interesting material in the archive.

Twitter This past year, the Library of Congress acquired a database of all public messages posted on Twitter, a micro-blogging site where each message is less than 140 characters. Searching, browsing, and accessing information in this database is a completely open problem. For instance, how can we browse through millions of tweets, the name for these short 140 character messages, about a topic such as the confirmation of Supreme Court justice Sonia Sotomayor? And what information

[4]Collected from http://en.wikipedia.org/wiki/List_of_countries_by_English-speaking_population on 1 September 2010.

are people interested in finding in Twitter archives? These are not yet mathematical tractable problems, but beg for new mathematical models designed for Twitter. Recent research on Twitter has identified many fascinating properties among the activity of users. See refs. [19, 21, 28] for basic statistics of behavior on Twitter, see [30] for a discussion of whether Twitter behaves more like a social network or news site—they conclude its more news like; and finally see [44] for a way to model authority among Twitter users.

8.3 Sources of data

Much of this paper was focused on using open data to help enrich the searching experience on the Library of Congress's proprietary data. This may have left readers wondering how they can contribute. As we mentioned before, the success of open data has produced a flood of freely available datasets. Here are some of our favorites:

- The Library of Congress subject headings—now freely available http://id.loc.gov/authorities
- Rameau—the French national library subject headings http://www.cs.vu.nl/STITCH/rameau/
- Freebase—a large collection of structured and semi-structured information http://freebase.com
- Open library—metadata about books http://openlibrary.org/

Each dataset provides the data in bulk form. This makes it straightforward to interpret the data. Freebase is composed of many small collections. Designing search and browse techniques for these individual collections is somewhat akin to the proprietary metadata from the Library.

Another possibility we recommend is to contact those in charge of your university or national library. In our experience, these institutions are eager for new ideas and approaches. Taking this approach, however, requires some patience while learning more about the subject of library and information science—the historical home for the study of information organization and access.

Acknowledgements We are indebted to the wonderful people at the Library of Congress, the World Digital Library, and the National Digital Information Infrastructure Preservation Program for telling us about their problems with digital stewardship and patiently helping us throughout our education on their problems. In particular, we'd like to specially acknowledge: Laura Campbell, George Coulbourne, Beth Dulabahn, Jane Mandelbaum, Barbara Tillett, as well as the Librarian of Congress: James Billington.

We'd also like to acknowledge Mohsen Bayati, who helped us formulate a scalable algorithm for the network alignment problem, and Les Fletcher, who frequently provided useful advice.

References

1. Adai, A.T., Date, S.V., Wieland, S., Marcotte, E.M.: LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. J. Mol. Biol. **340**(1), 179–190 (2004). doi:10.1016/j.jmb.2004.04.047
2. Bayati, M., Gerritsen, M., Gleich, D.F., Saberi, A., Wang, Y.: Algorithms for large, sparse network alignment problems. In: Proceedings of the 9th IEEE International Conference on Data Mining, pp. 705–710 (2009). doi:10.1109/ICDM.2009.135
3. Blondel, V.D., Gajardo, A., Heymans, M., Senellart, P., Dooren, P.V.: A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM Rev. **46**(4), 647–666 (2004). doi:10.1137/S0036144502415960

4. Chew, P.A., Bader, B.W., Kolda, T.G., Abdelali, A.: Cross-language information retrieval using PARAFAC2. In: Berkhin, P., Caruana, R., Wu, X., Gaffney, S. (eds.) Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD, pp. 143–152. Association for Computing Machinery, ACM Press, San Jose (2007). doi:10.1145/1281192.1281211
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. J. Pattern Recognit. Artif. Intell. **18**(3), 265–298 (2004). doi:10.1142/S0218001404003228
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002). http://gate.ac.uk/sale/acl02/acl-main.pdf
7. de Groat, G.: Future directions in metadata remediation for metadata aggregators. Tech. rep., Digital Library Federation (2009)
8. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990). doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9
9. Ehrig, M., Staab, S.: QOM—quick ontology mapping. In: Third International Semantic Web Conference. LNCS, vol. 3298, pp. 683–697 (2004). doi:10.1007/b102467
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 363–370. Association for Computational Linguistics, Morristown (2005). doi:10.3115/1219840.1219885
11. Fraikin, C., Nesterov, Y., Dooren, P.V.: A gradient-type algorithm optimizing the coupling between matrices. Linear Algebra Appl. **429**(5–6), 1229–1242 (2008). doi:10.1016/j.laa.2007.10.015
12. Fraikin, C., Nesterov, Y., Van Dooren, P.: Optimizing the coupling between two isometric projections of matrices. SIAM J. Matrix Anal. Appl. **30**(1), 324–345 (2008). doi:10.1137/050643878
13. Göbel, F., Jagers, A.A.: Random walks on graphs. Stoch. Process. Appl. **2**(4), 311–336 (1974). doi:10.1016/0304-4149(74)90001-5
14. Halevy, A., Norvig, P., Pereira, F.: The unreasonable effectiveness of data. IEEE Intell. Syst. **24**(2), 8–12 (2009). doi:10.1109/MIS.2009.36
15. Harshman, R.A.: PARAFAC2: Mathematical and technical notes. UCLA Work. Pap. Phon. **22**, 30–44 (1972)
16. Heer, J., Bostock, M.: Crowdsourcing graphical perception: using Mechanical Turk to assess visualization design. In: CHI '10: Proceedings of the 28th International Conference on Human Factors in Computing Systems, pp. 203–212. ACM, New York (2010). doi:10.1145/1753326.1753357
17. Higham, N.J.: Handbook of Writing for the Mathematical Sciences. SIAM, Philadelphia (1998)
18. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. Data Knowl. Eng. **67**(1), 140–160 (2008). doi:10.1016/j.datak.2008.06.003
19. Huberman, B.A., Romero, D.M., Wu, F.: Social networks that matter: Twitter under the microscope. First Monday **14**(1), Online (2008). URL http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2317/2063
20. Java, A.: Twitter social network analysis. UMBC ebquity blog (2007). URL http://ebiquity.umbc.edu/blogger/2007/04/19/twitter-social-network-analysis/
21. Java, A., Song, X., Finin, T., Tseng, B.: Why we Twitter: understanding microblogging usage and communities. In: WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56–65. ACM, New York (2007). doi:10.1145/1348549.1348556
22. Jia, Y., Hoberock, J., Garland, M., Hart, J.: On the visualization of social and other scale-free networks. IEEE Trans. Vis. Comput. Graph. **41**(6), 1285–1292 (2008). doi:10.1109/TVCG.2008.151
23. Karypis, G.: CLUTO—a clustering toolkit. Tech. Rep. 02-017, University of Minnesota, Department of Computer. Science (2002). URL http://glaros.dtc.umn.edu/gkhome/views/cluto/
24. Katz, S.M.: Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans. Acoust. Speech Signal Process. **35**(3), 400–401 (1987)
25. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with Mechanical Turk. In: CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, pp. 453–456. ACM, New York (2008). doi:10.1145/1357054.1357127
26. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95, vol. 1, pp. 181–184 (1995). doi:10.1109/ICASSP.1995.479394

27. Körner, C., Benz, D., Hotho, A., Strohmaier, M., Stumme, G.: Stop thinking, start tagging: tag semantics emerge from collaborative verbosity. In: WWW '10: Proceedings of the 19th International Conference on World Wide Web, pp. 521–530. ACM, New York (2010). doi:10.1145/1772690.1772744
28. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about Twitter. In: WOSP '08: Proceedings of the First Workshop on Online Social Networks, pp. 19–24. ACM, New York (2008). doi:10.1145/1397735.1397741
29. Kuny, T.: A digital dark ages? Challenges in the preservation of electronic information. In: 63rd International Federation of Library Associations and Institutions Council and General Conference (IFLA1997) (1997). URL http://ifla.queenslibrary.org/iv/ifla63/63kuny1.pdf
30. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media. In: WWW '10: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600. ACM, New York (2010). doi:10.1145/1772690.1772751
31. Levy, S.: How Google's algorithm rules the web. Wired Mag. **18**(3) (2010). http://www.wired.com/magazine/2010/02/ff_google_algorithm/all/1
32. McCallum, A.: Information extraction: Distilling structured data from unstructured text. Queue **3**(9), 48–57 (2005). doi:10.1145/1105664.1105679
33. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th International Conference on Data Engineering, p. 117. IEEE Computer Society, San Jose (2002)
34. Ninove, L.: Dominant vectors of nonnegative matrices: Application to information extraction in large graphs. Ph.D. thesis, Université Catholique de Louvain (2008)
35. Rafiei, D., Curial, S.: Effectively visualizing large networks through sampling. Vis. Conf., IEEE **0**, 48 (2005). doi:10.1109/VIS.2005.25
36. Rajaraman, A.: More data usually beats better algorithms. Datawocky Blog (2008). URL http://anand.typepad.com/datawocky/2008/03/more-data-usual.html
37. Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers? Shifting demographics in Mechanical Turk. In: CHI EA '10: Proceedings of the 28th International Conference Extended Abstracts on Human Factors in Computing Systems, pp. 2863–2872. ACM, New York (2010). doi:10.1145/1753846.1753873
38. Seidman, S.B.: Network structure and minimum degree. Soc. Netw. **5**(3), 269–287 (1983). doi:10.1016/0378-8733(83)90028-X
39. Silberman, M.S., Ross, J., Irani, L., Tomlinson, B.: Sellers' problems in human computation markets. In: HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation, pp. 18–21. ACM, New York (2010). doi:10.1145/1837885.1837891
40. Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday (2005)
41. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions, pp. 33–40. Association for Computational Linguistics, Morristown (2003). doi:10.3115/1119282.1119287
42. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 173–180. Association for Computational Linguistics, Morristown (2003). doi:10.3115/1073445.1073478
43. Various: The MARC Standard. URL http://www.loc.gov/marc (2007). Accessed on 17 September 2007
44. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: WSDM '10: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 261–270. ACM, New York (2010). doi:10.1145/1718487.1718520
45. Yee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 401–408. ACM, New York (2003). doi:10.1145/642611.642681
46. Yeh, E., Ramage, D., Manning, C.D., Agirre, E., Soroa, A.: Wikiwalk: random walks on wikipedia for semantic relatedness. In: TextGraphs-4: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, pp. 41–49. Association for Computational Linguistics, Morristown (2009)