Detecting and Modeling Local Text Reuse

David A. Smith
College of Computer and
Information Science
Northeastern University
Boston, MA, U.S.A.
dasmith@ccs.neu.edu

Ryan Cordell
Elizabeth Maddock Dillon
English Department
Northeastern University
Boston, MA, U.S.A.
{r.cordell,e.dillon}@neu.edu

Nick Stramp
John Wilkerson
Political Science Department
University of Washington
Seattle, WA, U.S.A.
{stramp,jwilker}@uw.edu

ABSTRACT

Texts propagate through many social networks and provide evidence for their structure. We describe and evaluate efficient algorithms for detecting clusters of reused passages embedded within longer documents in large collections. We apply these techniques to two case studies: analyzing the culture of free reprinting in the nineteenth-century United States and the development of bills into legislation in the U.S. Congress. Using these divergent case studies, we evaluate both the efficiency of the approximate local text reuse detection methods and the accuracy of the results. These techniques allow us to explore how ideas spread, which ideas spread, and which subgroups shared ideas.

1. INTRODUCTION

Many studies of social networks and interactions use surveys and observational data on nodes and links. This metadata is often augmented by the language the members of a social network use to interact with each other—e.g., the text of email messages within a company or of posts and tweets exchanged among online social network participants. In some cases, however, we cannot directly observe network links, or even get a census of network nodes, and yet we can still observe text that provides evidence for social interactions.

A particularly useful form of evidence comes from copying and quotation of text by different actors in a social network. The social ties indicated by text reuse may be overt, as with acknowledged quotations of earlier scholarly work, or unacknowledged, as with legislators' adoption, when drafting bills, of language from interest groups' policy documents. In aggregate, text reuse patterns can provide evidence for direct or indirect contact between different social actors. Examination of the reused passages can provide evidence for the shared interests of groups. At a finer level of analysis, the temporal and spatial dynamics of text reuse can provide evidence for individual links and propagation patterns in networks [1, 2].

Our approach to local text reuse detection and clustering is motivated by two case studies: (1) analyzing a corpus of nineteenth century U.S. newspapers to investigate the "culture of reprinting" that existed before effective copyright enforcement and wire services and (2) analyzing the text of bills proposed, amended, and passed in the U.S. Congress in order to determine how policy ideas progress and change in legislative bodies. While this paper focuses on the textual evidence alone, a more general setup could employ both text and information about some *related* network. Namata et al. [3], for instance, considered the task of inferring corporate hierarchies from the network of email communications. In the case study on 19c newspapers, we might use the railroad and telegraph networks as additional evidence for how news and culture propagated; in the congressional case study, voting records, committee assignments, interest group endorsements, and staff employment could provide similar side information about the mechanism for influence.

Researchers in NLP and information retrieval have often employed text reuse detection to remove duplicates from web crawls and search results and to detect plagiarism in documents and source code. These methods can achieve quite high precision when the majority of documents' contents are implicated in reuse. Performance is often tuned for ad-hoc retrieval, with one or a few query documents (e.g., student submissions) being run against a large corpus of potential matches. We are interested, however, in the overall patterns of text reuse and focus here on a different task: near-neighbor search among all document pairs for overlapping passages. For a web crawler, lower recall simply leads to a larger index; in our case, low recall on the text reuse task can lead to inaccurate output.

At a high level, the method proposed in this paper (1) finds pairs of documents likely to contain substantial overlap, (2) performs an alignment to find passages of local similarity even if the rest of the documents are unrelated, and (3) uses that pairwise passage information to infer larger clusters of text reuse. As we explain in more detail below, there are two primary sources of error in this process, in addition to the appearance of resemblance arising by chance that affects other work on sequence similarity from NLP to bioinformatics. First, since our collections contain multiple documents from the same source, precision can be hurt by often substantial amounts of intra-source text reuse, which are not as important when looking for evidence of interaction among sources. Second, since we are often interested in mapping networks in historical data, text that has been poorly transcribed by optical character recognition can de-

978-1-4799-5569-5/14/$31.00 ©2014 IEEE.

press recall.

Our approach, like many others, involves "shingling"—evaluating document pairs by the intersection of their n-gram features—, but these considerations require further refinements to optimize effectiveness and efficiency. In particular, we employ hashing for space-efficient indexing of repeated n-grams and the use of overlapping n-grams to prune dynamic programming for local alignment.

Before describing methods for detecting and clustering passages of local text reuse in §3, we provide background on our two case studies in order to motivate some of our design choices (§2). In §4.1, we describe experimental evaluations of the effectiveness and efficiency of various text reuse detection approaches and of the ability of text reuse approaches to detect actual policy borrowing in legislative bills. Finally, we perform two exploratory analyses: of the temporal dynamics of newspaper reprinting and of the classification of different kinds of reuse in bills (§5).

2. CASE STUDIES IN TEXT REUSE

The case studies in this paper, which form the basis for our experimental evaluation below, involve two fairly divergent domains: the informational and literary ecology of the nineteenth-century United States and the process of bill introduction and modification in the recent U.S. Congress. In this section, we describe the processes of text reuse that operate in those domains and the datasets we use to investigate them. The data range from noisy OCR from hundreds of archives covering texts over a twenty-year period with no divisions into articles (19c newspapers) to born-digital documents from a recent two-year period with machine-readable section breaks (Congressional bills).

The methods we discuss in this paper could also be applied, of course, to other domains. Most closely related to previous work on plagiarism detection is the investigation in historical and literary studies of the sources that influenced a text, sometimes referred to as *source criticism.*

2.1 Tracking Viral Texts in 19c Newspapers

In *American Literature and the Culture of Reprinting*, McGill [4] argues that American literary culture in the nineteenth century was shaped by the widespread practice of reprinting stories and poems, usually without authorial permission or even knowledge, in newspapers, magazines, and books. Without substantial copyright enforcement, texts circulated promiscuously through the print market and were often revised by editors during the process. These "viral" texts—be they news stories, short fiction, or poetry—are much more than historical curiosities. The texts that editors chose to pass on are useful barometers of what was exciting or important to readers during the period, and thus offer significant insight into the priorities and concerns of the culture. Nineteenth-century U.S. newspapers were usually associated with a particular political party, religious denomination, or social cause (e.g., temperance or abolition). Mapping the specific locations and venues in which varied texts circulated would therefore allow us to answer questions about how reprinting and the public sphere in general were affected by geography, communication and transportation networks, and social, political, and religious affinities.

To study the reprint culture of this period, we crawled the online newspaper archives of the Library of Congress's *Chronicling America* project (chroniclingamerica.loc.gov). Since the Chronicling America project aggregates state-level digitization efforts, there are some significant gaps: e.g., there are no newspapers from Massachusetts, which played a not insubstantial role in the literary culture of the period. While we continue to collect data from other sources in order to improve our network analysis, the current dataset remains a useful, and open, testbed for text reuse detection and analysis of overall trends.

Another difficulty with this collection is that it consists of the OCR'd text of newspaper issues without any marking of article breaks, headlines, or other structure. The local alignment methods described in §3.2 are designed not only to mitigate this problem, but also to deal with partial reprinting. One newspaper issue, for instance, might reprint chapters 4 and 5 of a Thackeray novel while another issue prints only chapter 5.

Since our goal is to detect texts that spread from one venue to another, we are not interested in texts that were reprinted frequently in the same newspaper, or *series*, to use the cataloguing term. This includes material such as mastheads and manifestos and also the large number of advertisements that recur week after week in the same newspaper.

2.2 Tracking Policy Ideas in Bills

At both the U.S. national and state levels, legislators can introduce as many bills as they prefer. In a typical two year session of the U.S. Congress, more than 10,000 public bills will be introduced (or about 19 bills per member on average). Some of these bill will propose very significant policy changes, such as the recently enacted Patient Protection and Affordable Care Act, while others will name post offices or require the U.S. Mint to produce a commemorative coin.

Only a small percentage (3-4%) of these 10,000 bills will ultimately be enacted into law and scholars have long been interested in what differentiates the small number of bills that survive the legislative gauntlet from those that do not. As valuable as this research is, it is limited by the fact that the focus is on the bill. Bills are "vehicles." An introduced bill contains legislative language, but the version that becomes law typically contains different legislative language [5]. In some cases, the final version is not much different from the introduced version; in other cases, the differences are substantial. In the 111th Congress (2009–2010), HR 3590 as introduced was 6 pages long and titled the *Service Members Home Ownership Tax Act of 2009.* As enacted, the same bill was titled the *Patient Protection and Affordable Care Act* (otherwise known as Obamacare) and was 906 pages long!

Our objective is to develop a systematic approach to modeling the incorporation of policy ideas *in* bills. Some policy ideas may be pulled from earlier, unpassed legislation or included from current proposed bills in order to gather their sponsors' support. We operationalize the notion of a policy idea as a single bill section. Further, it is statutory convention that "[e]ach section [of Acts and Resolutions] shall be numbered, and shall contain, as nearly as may be possible, a *single* proposition of enactment (emphasis added)."[1] Multiple sections may make up a single policy idea, but a section-based focus is unlikely to aggregate multiple ideas.

Congressional bill texts are available in digital form from 1989 to the present. Although there are some complications,

[1] Chapter 2, Section 104 of the U.S. Code, www.law.cornell.edu/uscode/.

each printed version of a bill is also available. Thus, we can compare not only different bills, but different versions of the same bill as it moves through the lawmaking process. As with the different issues of the same newspaper series discussed above, we are not directly interested in patterns of text reuse among different versions of the same bill, but in reuse among versions of different bills.

3. TEXT REUSE DETECTION

As noted above, we are interested in detecting passages of text reuse (individual articles) that comprise a very small fraction of the containing documents (newspaper issues). Using the terminology of biological sequence alignment, we are interested in *local alignments* between documents. Henzinger [6] provides a good overview of text reuse detection research and provides an empirical evaluation of the two primary methods—n-gram shingling and locality-sensitive hashing (LSH)—on near-duplicate webpage detection tasks. The need for local alignments makes LSH less practical without performing a large number of sliding-window matches.[2] We therefore base our approach on n-gram shingling.

Several attributes distinguish the present approach from previous work:

- The boundaries of the reused passages are not known, in contrast to near-duplicate document detection and to work on "meme tracking" that takes text between quotation marks as the unit of reuse [1, 8].

- Also in contrast to this work on the contemporary news cycle and blogosphere, we are interested both in texts that are reprinted within a few days and after many years. We thus cannot exclude potentially matching documents for being far removed in time.

- We are looking for reuse of substantial amounts of text, on the order of 100 words or more, in contrast to work on detecting shorter quotations [9, 10, 11, 2].

- We wish to compute all nearest neighbor passages, instead of running a small number of query documents against a corpus.

- Text reuse that occurs only among documents from the same "source" should be excluded. Henzinger [6] notes, for instance, that many of the errors in near-duplicate webpage detection arose from false matches among documents from the same website that shared boilerplate navigational elements.

- We require methods that are robust to noisy OCR transcripts. While we could adopt a pipelined approach and perform OCR correction before text reuse detection, it is worth noting that a promising source of data for OCR correction are the clusters of similar passages found in other documents.

In common with work in duplicate detection, we start the search for reused passages by first finding likely document pairs. For each document pair returned by this first step, we then identify a passage embedded within each document that is a good local match for the other. We then greedily aggregate this pairwise data into large clusters of reused text passages. We now describe the details of each step in turn.

3.1 Search for Candidate Document Pairs

We begin with a scan for document pairs likely to contain reprinted passages. We use "shingling", which represents documents by an unordered collection of its n-grams, to provide features for document similarity computations. We balance recall and precision by adjusting the size of the shingles and the proportion of the shingles that are used to calculate document similarity. After determining which features will be used to represent documents, we build an index of the repeated n-grams and then extract candidate document pairs from this index. Table 1 shows the parameters used in our approach.

3.1.1 N-gram Document Representations

The choice of document representation affects the precision and recall, as well as the time and space complexity of a text reuse detection system. In our experiments, we adopt the popular choice of n-grams, contiguous subsequences of n words. Very similar passages of sufficient length will share many long subsequences, but long n-grams are less robust to textual variation and OCR errors. As we see in the experiments below (§4.1), values of n between 5 and 7 provide a reasonable tradeoff between accuracy and efficiency for the newspaper corpus while longer n-grams work well on the cleaner legislative corpus.

In previous work on detecting short quotes or "memes", short contiguous n-grams have been proven to be quite effective. Suen et al. [8], for instance, use 4-grams to identify similar quotes in newspaper articles and blogs. Many fixed phrases, however, will also be detected by that approach. For instance, two documents are not much more likely to be derived from the same source if they both include the 5-grams "it is no accident that" and "in the final analysis it". We will see how to mitigate this problem by upper bounding the document frequency of the n-grams we use.

In addition to standard contiguous n-grams, therefore, we also explored the use of *skip n-grams*, or non-contiguous subsequences of the document. Skip n-grams allow us to look at subsequences of words that fall in more widely separate parts of the document. We parameterized our skip n-grams by n, the number of terms included; g, the minimum gap between terms; and w, the maximum number of terms covered by the skip n-gram. A contiguous 5-gram would thus be n=5 g=1 w=5. In this paper, we confine ourselves to skip bigrams (n=2) with width up to 105.

3.1.2 Downsampling Document Features

Since duplicate texts will share many matching n-grams, many systems use only a small fraction of the n-grams to represent each document.

One of the most popular downsampling techniques is 0 mod p [12]. When indexing a document, this technique hashes each n-gram to an integer and then determines whether the hash value is divisible by some integer p. Since instances of the same n-gram in all documents hash to the same value, the algorithm does not need to maintain a dictionary of the downsampled features. This also permits easy parallelization since the hash functions can be computed independently in each batch. Many duplicate detection systems for the web use p = 50 or above, which drastically reduces index sizes

[2] While there is some work on embedding string edit distances for LSH, it applies to measures of *global* alignment such as Hamming distance and cyclic shifts [7].

[6]. This downsampling technique is less effective for local as opposed to global text reuse [10] and can also hurt recall for noisily OCR'd documents, as we see below.

For bigrams, we also adopt filtering on stopwords. While indexes of longer n-grams are not dominated by stopword-only entries, pairs of stopwords can increase the size of a bigram index significantly. Due to the prevalence of OCR errors in the newspaper dataset, we treat all words of four or fewer characters as stopwords. Also on the stoplist are the terms with the highest document frequency that make up 2/3 of the tokens in the corpus. On the newspaper collection, leaving out the four-character-or-less terms, this amounted to 1440 stopwords. We ignore any bigram with at least one stopword. This filter proved to be too stringent with longer n-grams, though some variants might be successful.

3.1.3 *Efficient N-gram Indexing*

The next step is to build for each n-gram feature an inverted index of the documents where it appears. As in other duplicate detection and text reuse applications, we are only interested in the n-grams shared by two or more documents. The index, therefore, does not need to contain entries for the n-grams that occur only once. We use the two-pass space-efficient algorithm described in Huston et al. [13], which, empirically, is very efficient on large collections. In a first pass, n-grams are hashed into a fixed number of bins. On the second pass, n-grams that hash to bins with one occupant can be discarded; other postings are passed through. Due to hash collisions, there may still be a small number of singleton n-grams that reach this stage. These singletons are filtered out as the index is written.

In building an index of n-grams, an index of (n-1)-grams can also provide a useful filter. No 5-gram, for example, can occur twice unless its constituent 4-grams occur at least twice. We do not use this optimization in our experiments; in practice, n-gram indexing is less expensive than the later steps.

3.1.4 *Extracting and Ranking Candidate Pairs*

Once we have an inverted index of the documents that contain each (skip) n-gram, we use it to generate and rank document pairs that are candidates for containing reprinted texts. Each entry, or *posting list*, in the index may be viewed as a set of pairs (d_i, p_i) that record the document identifier and position in that document of that n-gram.

Once we have a posting list of documents containing each distinct n-gram, we output all pairs of documents in each list. We suppress repeated n-grams that appear in different issues of the same newspaper. These repetitions often occur in editorial boilerplate or advertisements, which, while interesting, are outside the scope of this project. We also suppress n-grams that generate more than $\binom{u}{2}$ pairs, where u is a parameter.[3] These frequent n-grams are likely to be common fixed phrases. Filtering terms with high document frequency has led to significant speed increases with small loss in accuracy in other document similarity work [14]. We then sort the list of repeated n-grams by document pair, which allows us to assign a score to each pair based on the number of overlapping n-grams and the distinctiveness of those n-grams. When downsampling n-grams with 0 mod p, we use the entire ranked list; otherwise, we confine our evaluation to document pairs with 5 or more shared n-grams.

Table 1: Parameters for text reuse detection

n	n-gram order
w	maximum width of skip n-grams
g	minimum gap of skip n-grams
p	n-grams downsampled with $0 \mod p$
u	maximum distinct series in the posting list

3.2 Local Document Alignment

The initial pass returns a large ranked list of candidate document pairs, but it ignores the order of the n-grams as they occur in each document. We therefore employ local alignment techniques to find compact passages with the highest probability of matching. The goal of this alignment is to increase the precision of the detected document pairs while maintaining high recall. Due to the high rate of OCR errors, many n-grams in matching articles will contain slight differences.

Unlike some partial duplicate detection techniques based on global alignment [15], we cannot expect all or even most of the articles in two newspaper issues, or the text in two books with a shared quotation, to align. Rather, as in some work on biological subsequence alignment [16], we are looking for regions of high overlap embedded within sequences that are otherwise unrelated. We therefore employ the Smith-Waterman dynamic programming algorithm with an affine gap penalty. We use the simplest form of a weight matrix common in the bioinformatics community: matching characters have a weight of 2, mismatches have -1, opening a gap is -5, and continuing a gap is -0.5. This use of model-based alignment also distinguishes this approach for other work, for detecting shorter quotations, that greedily expands areas of n-gram overlap [9, 11]. We do, however, prune the dynamic programming search by forcing the alignment to go through position pairs that contain a matching n-gram from the previous step, as long as the two n-grams are unique in their respective texts. Table 3 shows an example of two aligned sections from a House and Senate bill. Hyphens are inserted into each text to indicate that the other text contains additional material.

Even the exact Smith-Waterman algorithm, however, is an approximation to the problem we aim to solve. If, for instance, two separate articles from one newspaper issue were reprinted in another newspaper issue in the opposite order—or separated by a long span of unrelated matter—the local alignment algorithm would simply output the better-aligned article pair and ignore the other. Anecdotally, we only observed this phenomenon once in the newspaper collection, where two different parodies of the same poem were reprinted in the same issue. In any case, our approach can easily align different passages in the same document to passages in two other documents.

3.3 Passage Clustering

We now have a set of aligned passage pairs. We sort the passage pairs in descending order by length and perform greedy single-link clustering. If two passages in the same document overlap by 80%, they are taken to be the same passage for purposes of clustering; otherwise, they are treated

[3] The filter is parameterized this way because it is applied after removing document pairs in the same series.

as different passages. Given this placement of passages into equivalence classes, we can view this step as detecting connected components in the graph of aligned passages. The V equivalence classes form the vertices; the edges are the E pairwise connections between passages determined by the previous alignment step. When clustering in a single pass through the pairwise data, using Tarjan's disjoint set algorithm requires $O(V)$ space and $O(V + E \cdot \alpha(E, V))$ time, where $\alpha(m, n)$ is the inverse Ackermann function [17]. This function grows so slowly that the algorithm is effectively linear in the input size. Table 2 shows an example cluster.

4. EVALUATION

We did not start out with an annotated test set for text reuse detection. We nevertheless performed empirical evaluations in order to measure the effectiveness and efficiency of the end-to-end system and to tune various free parameters. When trying to efficiently detect candidate passage pairs to align, we can evaluate various approximate search techniques against the results we could achieve with more exhaustive techniques, as in our first set of experiments (§4.1). We then (§4.2) describe an experiment with labeling aligned section pairs in the bills corpus to evaluate the ability of text reuse analysis to find shared policy ideas.

4.1 Evaluating Reprint Detection

For the pre-Civil War period, which is our focus of interest with the newspaper collection, our corpus contains 1.6 billion words from 41,829 issues of 132 newspapers. The collection was created with OCR of middling accuracy, as can be seen in table 2. For the congressional bill task, we evaluated on the bills in the U.S. House of Representatives in the 111th congress (2009–10). This collection contains 32 million words from 85,525 sections of 8923 versions of 6562 distinct bills.

To evaluate the precision and recall of text reuse detection, we create a pseudo-relevant set of document pairs by pooling the results of several runs with different parameter settings. For each document pair found in the union of these runs, we observe the length, in matching characters, of the longest local alignment. (Using matching character length allows us to abstract somewhat from the precise cost matrix.) We can then observe how many aligned passages each method retrieves that are at least 50,000 character matches in length, at least 20,000 character matches in length, and so on. The candidate pairs are sorted by the number of overlapping n-grams; we measure the pseudo-recall at several length cutoffs. For each position in a ranked list of document pairs, we then measure the precision: what proportion of documents retrieved are in fact 50k, 20k, etc., in length? Since we wish to rank documents by the length of the aligned passages they contain, this is a reasonable metric. As a summary of these precision values, we employ the *average precision*: the mean of the precision at every rank position that contains an actually relevant document pair. One of the few earlier evaluations of local text reuse, by Seo and Croft [10], compared fingerprinting methods to a trigram baseline. Since their corpus contained short individual news articles, the extent of the reused passages was evaluated qualitatively rather than by alignment.

Table 4 shows the average precision of different parameter settings on the newspaper collection, ranked by the number of pairs each returns. If the pairwise document step returns a large number of pairs, we will have to perform a large number of more costly Smith-Waterman alignments. On this collection, a good tradeoff between space and speed is achieved by skip bigram features. In the best case, we look at bigrams where there is a gap of at least 95, and not more than 105, words between the first and second terms (n=2 u=100 w=105 g=95).

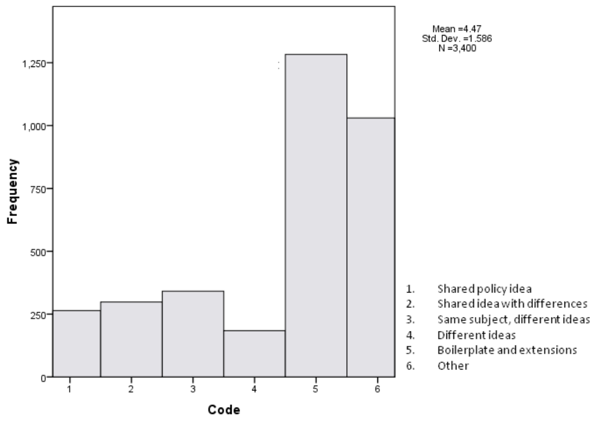

Figure 1: Histogram of human-labeled alignments of bill sections by category

Table 5 shows the average precision results on the congressional bills collection. Since this collection was created by OCRing modern documents or from born-digital documents, longer n-grams are much more effective than they are on the noisily OCR'd newspaper collection.

4.2 Classifying Policy Ideas

In investigating the legislative data, it became clear that bills share a lot of language about mundane but necessary things, such as defining terms, establishing effective dates, creating commissions, calling for reports, and adjusting for inflation, etc. We call these types of matches boilerplate. In addition, bill sections often share similar preambles that lack substance. Finally, two aligned texts can be similar in most respects, but differ in small but critical respects: both might propose similar medical education programs but for different professions (e.g. pediatrics vs. dentistry).

Such nuances mean that validation is essential. We constructed a gold standard human-labeled dataset to test the performance of predicting shared policy ideas. Our working definition of a policy idea is based on human judgment: the alignment must include text that provides a comprehensible description of a policy objective.

The initial sample includes 3400 of the 19,241 alignments related to the Affordable Care Act drawn from alignments that make up at least 7% of one or both sections (in the top 50% of cases). Note that we are labeling alignments, not bill sections, for similarity. Most of the alignments in our sample are boilerplate or preambles and junk. Instances of shared policy ideas make up about 16% of the sample, while cases where the aligned passages addressed different policy ideas make up slightly less (Figure 1).

The next step is to predict whether two texts share a policy idea. To do this we first train an SVM to predict the boilerplate cases using the 3400 human labeled cases. Next

Table 2: Example cluster from newspaper corpus. Only the beginnings of the text of the story, by popular temperance writer T. S. Arthur, are shown. In addition to the many obvious OCR errors, there are editorial changes, as well, such as omitting "Mr." in the third text or changing "backwards" to "backward" in the second two texts. (These changes were checked against the page images.) Such changes provide evidence for how texts propagate.

1841-04-17 Sunbury American	`soft answer ny t s arthur ill give him law to his hearts content the scoundrel said mr singleton walking backwards and forwards`
1847-02-05 Anti-slavery Bugle	`soft aasffch bv t s arthuit 1 4 ill give him law to his hearts coii ent fhe scoundrel said mr single on walking backwards and forwards`
1847-05-04 Somerset Herald	`soft answer by t s arthur ill ffive him inw to his hearts content the scoundrel said singleton walking bick ward and forward`
1847-07-22 Vermont Watchman	`soft answer ey t s arthur ill give him law to his hearts content the scoundrep said singleton walking backward and forward`

Table 3: A Local Alignment Example with a passage from S 1244 (Breastfeeding Promotion Act of 2009) on the left and the PPACA (HR 3590) on the right.

`ing mothers a in general section`	`ing mothers--------- section 7`
`7 of the fair labor standards`	`of the fair labor standards act`
`act----- 29 usc 207 is amended by`	`of 1938 29 usc 207 is amended by`
`adding at the end the following`	`adding at the end the following`
`r 1 an employer shall provide--`	`r 1 an employer shall provide a`
`- reasonable break time for an`	`a reasonable break time for an`
`employee to express breast milk`	`employee to express breast milk`
`for her nursing child for 1 year`	`for her nursing child for 1 year`
`after the childs birth each time`	`after the childs birth each time`
`such employee has need to express`	`such employee has need to express`
`the milk the employer shall make`	`the milk and ----------------`
`reasonable efforts to provide`	`----------------b a place other`
`a place other than a bathroom`	`than a bathroom that is shielded`
`that is shielded from view and`	`from view and free from intrusion`
`free from intrusion from cowork-`	`from coworkers and the public`
`ers and the public which may be`	`which may be used by an employee`
`used by an employee to express`	`to express breast milk 2 an em-`
`breast milk- an employer shall`	`ployer shall not be required to`
`not be required to compensate`	`compensate an employee receiving`
`an employee--------------------`	`reasonable break time under para-`
`------------- for any work time`	`graph 1 for any work time spent`
`spent for such purpose 2 for pur-`	`for such purpose 3 ------------`
`poses of this subsection the term`	`-------------an employer ---that`
`employer means an employ`	`employ`

Table 4: Average precision on detecting reprinted passages of different minimum lengths in the newspaper collection, measured in the number of *matching* characters. Table 1 explains the parameter symbols.

		Average precision on passages at least					
Method	Pairs	50k	20k	10k	5k	2k	1k
n=10 u=100	3,342,162	0.18	0.09	0.10	0.10	0.24	0.31
n=7 u=50	3,579,992	0.22	0.18	0.20	0.20	0.22	0.30
n=2 u=50 w=55 g=45	4,297,764	0.32	0.30	0.28	0.27	0.21	0.27
n=6 u=50	4,433,792	0.21	0.20	0.22	0.21	0.21	0.31
n=7 u=100	5,971,269	0.20	0.11	0.12	0.14	0.30	0.44
n=5 u=50	6,443,100	0.21	0.22	0.25	0.22	0.20	0.31
n=2 u=100 w=105 g=95	7,512,442	0.34	0.30	0.25	0.29	0.40	0.47
n=2 u=100 w=55 g=45	9,756,985	0.31	0.26	0.21	0.24	0.32	0.45
n=2 u=100 w=25 g=15	12,798,056	0.28	0.24	0.19	0.21	0.26	0.39
n=5 u=100	15,258,185	0.19	0.15	0.15	0.17	0.30	0.47
n=4 u=50	17,954,922	0.19	0.23	0.27	0.23	0.18	0.28
n=5 u=100 p=10	43,701,236	0.19	0.15	0.15	0.16	0.26	0.35
n=4 u=100	71,263,521	0.16	0.17	0.17	0.18	0.26	0.42
n=5 u=100 p=5	77,771,798	0.18	0.15	0.15	0.17	0.28	0.41

Table 5: Average precision on the bills corpus.

		Average precision on passages at least					
Method	Pairs	50k	20k	10k	5k	2k	1k
n=7 u=50	571,874	0.45	0.40	0.38	0.52	0.69	0.76
n=5 u=50	806,158	0.47	0.39	0.37	0.51	0.68	0.74
n=10 u=100	856,039	0.42	0.37	0.39	0.55	0.72	0.79
n=7 u=100	1,365,547	0.42	0.35	0.38	0.54	0.71	0.79
n=2 u=100 w=105 g=95	1,995,208	0.36	0.56	0.46	0.56	0.67	0.59
n=5 u=100	2,042,445	0.44	0.35	0.37	0.52	0.69	0.77
n=2 u=100 w=55 g=45	2,389,442	0.38	0.48	0.40	0.51	0.66	0.66
n=2 u=100 w=25 g=15	2,769,449	0.37	0.43	0.36	0.48	0.63	0.67
n=5 u=100 p=10	4,758,741	0.43	0.34	0.37	0.52	0.68	0.76
n=5 u=100 p=5	8,314,137	0.43	0.35	0.38	0.52	0.69	0.76

Table 6: Detecting policy ideas shared among bill sections. A first stage classifier removes non-substantive "boilerplate" language. Results from 20-fold cross validation (2900 train, 500 test).

	Avg. Acc. [%]	95% Conf.
Predicting Boilerplate language (SVM):		
Correctly Predicted	85.6	(82.4–88.2)
Sensitivity (true positives)	76.5	(70.8–82.3)
Specificity (true negatives)	91.1	(87.8–94.0)
Predicting Shared Policy (Logistic Reg.):		
Correctly Predicted	87.4	(84.6-89.8)
Sensitivity (true positives)	69.2	(59.0–78.4)
Specificity (true negatives)	90.9	(87.9–93.5)
Predicting Shared Policy w/o Boilerplate (SVM + LR):		
Correctly Predicted	92.0	(89.6–94.0)
Sensitivity (true positives)	65.0	(55.1–74.3)
Specificity (true negatives)	97.3	(95.4–98.8)

we partition the remaining human labeled cases into those containing shared policy idea and those without. We then utilize 20-fold cross validation to divide cases into training and testing sets and predict shared policy ideas via logistic regression (including and then excluding boilerplate cases). The only independent variable in the logistic regression is the Smith-Waterman alignment score.

Table 6 reports the results of this experiment. The last set of results are for the most complete method. The overall prediction rate is 92%. Recall is 97.3%, which means that a true shared policy idea is missed only 2.7% of the time. The false positive rate is higher (about 31%), but because there are far fewer predicted policy ideas to review, false negatives are of less concern than false positives. Upon closer inspection, many of the false positives were boilerplate cases that the initial SVM missed.

5. EXPLORATORY DATA ANALYSIS

Running these text reuse detection methods on the newspaper and bills collections, we produced initial clusters of passages. In this section, we present exploratory analyses of different genres and patterns of appearance of texts uncovered in both corpora.

5.1 Geographic and Network Analysis

Looking beyond the details of individual texts, we can use our large and growing index of "viral" texts to begin modeling the systems that underlay nineteenth-century print culture. If we take reprinted texts as strong indicators of connection and influence among newspapers and among their editors, we can use our database of recirculated texts to map those connections, both geographically and as a network. Literary scholars and historians have in the past been limited in their analyses of print culture by the constraints of physical archives and human capacity. A lone scholar cannot read, much less make sense of, millions of newspaper pages. With the aid of computational linguistics tools and digitized corpora, however, we are working toward a large-scale, systemic understanding of how texts were valued and transmitted during this period.

Using geographic analysis, our uncovered print histories can be correlated with other spatial data, such as historical county boundaries, census reports, and transportation data, to construct individual and comparative models of geographic dispersal. Using only the broadest data point from the census—raw population—we can compare the potential audiences for different "viral" texts. For instance, John Greenleaf Whittier's agricultural hymn, "The Huskers," was reprinted 30 times within our data set, and 634,499 people lived within 5 miles of its sites of reprinting. By contrast, the Scottish poet Charles MacKay's religious poem, alternately titled "The Inquiry" and "The Resting Place," was reprinted 30 times with 5 miles of 609,709 people. There are a range of other interesting data points in the census reports that we plan to explore more fully as the project develops. In 1840, for instance, census takers were interested in the print industry, and they recorded the number of daily, weekly, and monthly newspapers in each county, as well as the number of magazines and other periodicals. These data will help us understand how the print culture reconstructed through recirculation data lines up (or does not line up) with 19c Americans' understanding of print culture.

Perhaps most compelling in our early spatial analyses has been the close correlation between the growth of reprinting networks and the growth of transportation technologies, particularly the railroad. As the rail network grows, connecting ever-smaller pockets of the country to the major hubs in the east, so to do the reprinted stories spread further and further outside major cities. Indeed, traffic seems to have moved in both directions, though unevenly. While the majority of the "viral" texts in our data set do seem to begin in major cities, such as New York, and spread from there to the south and west, a more significant minority of texts than we expected move in the opposite direction, beginning in smaller country newspapers and spreading from there into more metropolitan papers. In future, we plan to develop more robust techniques to reconstruct stemmas, or phylogenetic trees, of textual transmission.

Finally, visualizing our data as a network exposes po-

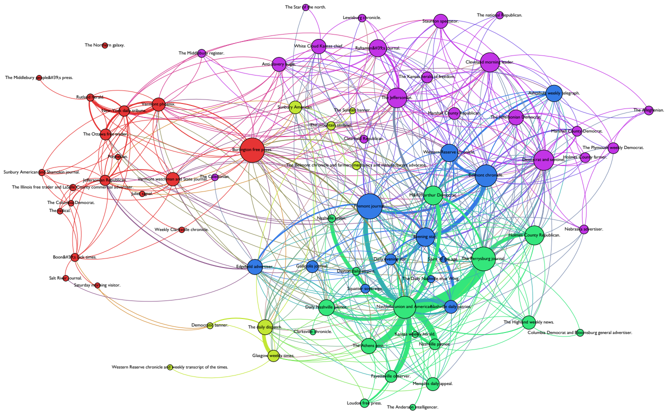

Figure 2: Visualization of the links among newspapers induced by shared reprinted texts. Colors indicate communities derived from graph modularity.

tential connections between newspapers beyond geography (figure 2). In these graphs, the nodes represent individual newspapers, while the edges are weighted based on how many texts two newspapers share. The communities of practice highlighted in these graphs often stretch across antebellum America. One community of reprinting partners, for instance, includes newspapers in Vermont, New York, Ohio, and Missouri, which was in the west of the United States during the period of our study. These connections suggest further research. When the network graphs suggested a close connection between the *Vermont Phoenix* (Brattleboro, Vermont) and the *Fremont Journal* (Fremont, Ohio), for instance, we investigated further, only to discover that the editors of these papers were brothers-in-law. These geographic and network modeling techniques, then, have pointed to new and relevant questions for humanities fields such as history and literary studies.

5.2 Reuse Dynamics: Reading Fast and Slow

For a reader of today, one of the most striking aspects of 19c U.S. newspapers is the mixture of what we would recognize as news items with other genres such as poetry, chapters of novels, jokes, and morally edifying anecdotes of vague origin. We explore how different genres may be distinguished not only by their content but also by the temporal pattern of their spread.

Some texts are reprinted frequently at first and then become less popular; others continue to attract interest from newspaper editors—and presumably readers—for years afterwards. Figure 3 shows the median lag time, in log number of days, between the first observed appearance and later reprints. The lag times are plotted by the year of a text's first occurrence. (We omit plots for 1856–60 due to boundary effects.) Most obviously, there are simply more texts (and more newspapers) as time goes on. In many years, there are two peaks in the frequency of lag time, at around one week and over 3 years. In 1846–7, for example, there is an increase in short lag-time, "fast" texts, due in part to coverage of the Mexican-American War. These fast texts later become more frequent. We suspect that the rise of wire services plays a role, though more research is needed.

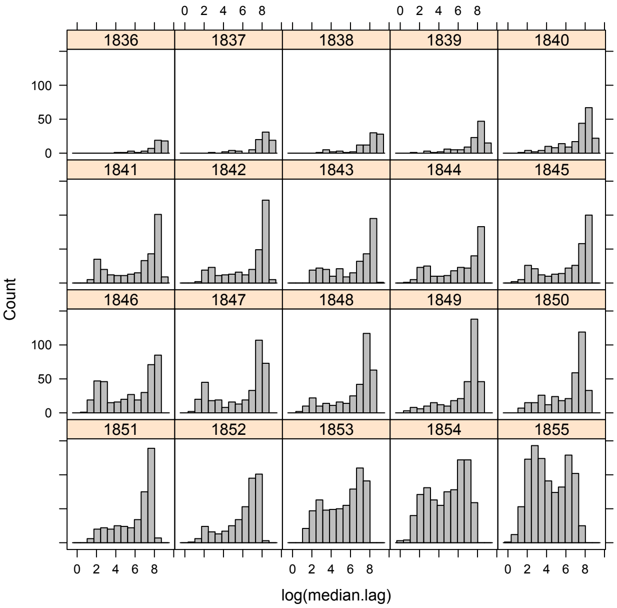

Figure 3: Median lag time, in log days, of reprints from first appearance, by year of first appearance. Many years show peaks around 2 (≈ 1 week) and 7 (≈ 3 years).

Not surprisingly, different kinds of texts travel fast and slow. We divided reprinted texts into a fast set, with median lag times under one year, and a slow set, with median lag times over five years, which gave approximately equal sets. We trained a logistic regression model to predict whether a text would be fast or slow. Features were all words in a text's first occurrence that (1) were five characters or more in length and (2) occurred five or more times in the corpus. The year of first occurrence was included as a categorical feature, to account for the different proportions in fast and slow texts over time. The training objective function was regularized with an L_1 (Lasso) penalty to achieve a sparse set of predictors [18].

Table 7 shows the top negative ("fast") and positive ("slow") coefficients. Highly correlated with fast textual propagation, for example, are terms related to the Mexican-American War, such as *Texas*, *Mexico*, and [Zachary] *Taylor*. Also interesting are *government* and *tariff*, *cases* and *corpse*. Slow texts focus on *love* and other affective terms, on *heaven* and interestingly *woman*. The year *1840* is also a useful predictor since fewer texts from that year were fast. With a random split between training and test, logistic regression achieved 73% accuracy on test. When earlier years (1840–1849) were used to predict a later one (1850), accuracy fell to 60%. Taking the cross product of year and word features only slightly diminished overfitting; more analysis of the rhetorical patterns of different genres should be helpful. Linear regression on the log median lag produced similar coefficients but fairly poor predictive accuracy.

5.3 Tracing the Development of the PPACA

To analyze the policy questions raised by text reuse, we looked at the results on all 19,241 candidate sections pairs

Table 7: Top features from model predicting the lag in reprinting. "Fast" texts have a median reprinting time of a year or less; "slow" texts have a median reprinting time of at least five years.

"Fast" texts		"Slow" texts	
texas	-0.700	love	0.459
mexico	-0.692	young	0.154
pills	-0.672	earth	0.138
taylor	-0.649	1840	0.129
tariff	-0.534	awoke	0.121
government	-0.502	fine	0.098
board	-0.463	silence	0.097
effect	-0.428	benevolence	0.078
whig	-0.426	soul	0.057
mate	-0.418	sweet	0.048
prices	-0.418	grant	0.035
goods	-0.416	hang	0.033
corpse	-0.387	behold	0.026
cases	-0.383	bright	0.025
general	-0.370	woman	0.020
public	-0.368	things	0.020
cure	-0.367	heaven	0.019

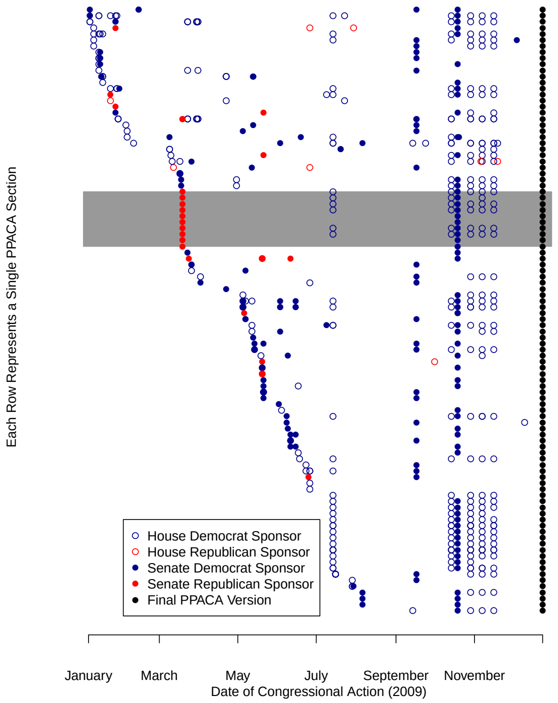

Figure 4: Sections of other bills sharing policy ideas with PPACA sections

that included a section from the Patient Protection and Affordable Care Act ("Obamacare"). This produced 1079 predicted cases of shared policy ideas. As a followup we tested the impact of lowering the "first pass" threshold from 10 grams to a more inclusive 5 grams. Doing so had no impact on these results. After inspecting and excluding false positives, the final dataset contained 1022 shared policy ideas. Some of these matches were with bills introduced after passage of the PPACA in December 2009. Excluding these cases further reduces the alignments of interest from 1022 to 844. To be clear, these are not necessarily 1022 distinct ideas. A single idea in the PPACA might match ideas found in different bills.

Figure 4 provides a chronological perspective to help illustrate the scope and origins of these policy ideas. Each row corresponds to a section of the PPACA. Due to space considerations the figure does not include unaligned PPACA sections or boilerplate sections. Each column is a point in time prior to the Senate's passage of the final version of the PPACA on December 24, 2009. The length of the vertical black line on the right represents the total number of PPACA sections. The colored dots in this case are sections from bills that align with that PPACA section. (Circles indicate sections from House bills, dots indicate sections from Senate bills.)

At the top are sections of HR 3590 (the PPACA) that are related chronologically to bills introduced early in the 111th Congress. Further down are PPACA sections whose origins appear to be more recent. Many are Republican sponsored. For example, the earliest bill linked to subtitle (B) of the PPACA relating to nursing home fraud prevention (the shaded portion) was S. 647, sponsored by Republican Senator Charles Grassley (R-IA). This is an example of a bill that the Congressional Research Service does not list as "related" to H.R. 3590.

In general, the considerations influencing which bills become law differ from those influencing which ideas progress. The Senate's decision to pass H.R. 3590 instead of another bill was a procedural move. Such moves appear to be fairly common, as are omnibus bills and cases where lawmakers exploit "must act" issues to advance unrelated policies. The majority party, particularly the majority leadership, has every incentive to claim credit for the passage of these bills. Who deserves credit for the ideas in these bills is an entirely different question. The progress of policy ideas should be influenced less by considerations of credit claiming, and more by considerations of problem solving and coalition building. As a result, patterns of effectiveness and inclusiveness at the level of the policy idea should be substantially different from patterns observed at the level of the bill.

Table 8 summarizes the sponsors of bills for the clearest cases of sections sharing policy ideas with the PPACA. The second half of the table excludes the 4 major markup vehicles. Republican-sponsored ideas were more likely to originate in Senate rather than House bills. More than one-fourth (27.8%) of the aligned Senate sections (excluding the major markup bills) can be traced to bills sponsored by Republican senators, compared to 10.7% to bills sponsored by House Republicans.

6. CONCLUSIONS

We have described efficient algorithms that exploit shingling, space-efficient hashed indexing, and local alignment

Table 8: “Inclusiveness” by chamber and party status

All aligned sections

s	House	Senate
Minority	2.8%	8.0%
Majority	97.9%	92.2%
N	468	376

Excluding markup bills

	House	Senate
Minority	10.7%	27.8%
Majority	89.3%	72.2%
N	122	108

to detect clusters of passages reprinted within large collections of OCR’d texts without a priori knowledge of which texts were repeated. Applied to collections of 19c newspapers and Congressional bills, these techniques allow us to explore how ideas spread, which ideas spread, and which subgroups shared ideas.

Acknowledgements

This work was supported in part by NEH Digital Humanities Start-Up Grant #HD-51728-13, by NSF grant #IIS-0910884, by a grant from the Andrew W. Mellon Foundation’s Scholarly Communications and Information Technology program, by Northeastern University’s Tier 1 grant program for interdisciplinary research, and by the NUlab for Texts, Maps, and Networks. Any views, findings, conclusions, or recommendations expressed do not necessarily reflect those of the NEH, NSF, or Mellon.

References

[1] J. Leskovec, L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle,” in *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2009, pp. 497–506.

[2] H. Ryu, M. Lease, and N. Woodward, “Finding and exploring memes in social media,” in *Hypertext*, 2012, pp. 295–304.

[3] G. M. Namata, S. Kok, and L. Getoor, “Collective graph identification,” in *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2011, pp. 87–95.

[4] M. L. McGill, *American Literature and the Culture of Reprinting, 1834–1853.* U. Penn. Press, 2003.

[5] E. S. Adler and J. D. Wilkerson, *Congress and the Politics of Problem Solving.* Cambridge University Press, 2012.

[6] M. Henzinger, “Finding near-duplicate web pages: A large-scale evaluation of algorithms,” in *ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, 2006, pp. 284–291.

[7] A. Andoni, A. Goldberger, A. McGregor, and E. Porat, “Homomorphic fingerprints under misalignments: Sketching edit and shift distances,” in *ACM Symp. on Theory of Computing (STOC)*, 2013, pp. 931–940.

[8] C. Suen, S. Huang, C. Eksombatchai, R. Sosič, and J. Leskovec, “NIFTY: A system for large scale information flow tracking and clustering,” in *Int. World Wide Web Conf.*, 2013, pp. 1237–1248.

[9] O. Kolak and B. N. Schilit, “Generating links by mining quotations,” in *Hypertext*, 2008, pp. 117–126.

[10] J. Seo and W. B. Croft, “Local text reuse detection,” in *ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, 2008, pp. 571–578.

[11] R. Horton, M. Olsen, and G. Roe, “Something borrowed: Sequence alignment and the identification of similar passages in large text collections,” *Digital Studies / Le champ numérique*, vol. 2, no. 1, 2010.

[12] Y. Bernstein and J. Zobel, “A scalable system for identifying co-derivative documents,” in *String Processing and Information Retrieval (SPIRE)*, 2004, pp. 55–67.

[13] S. Huston, A. Moffat, and W. B. Croft, “Efficient indexing of repeated n-grams,” in *ACM Web Search and Data Mining Conf. (WSDM)*, 2011, pp. 127–136.

[14] T. Elsayed, J. Lin, and D. W. Oard, “Pairwise document similarity in large collections with MapReduce,” in *ACL Short Papers*, 2008, pp. 265–268.

[15] I. Z. Yalniz, E. F. Can, and R. Manmatha, “Partial duplicate detection for large book collections,” in *ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2011, pp. 469–574.

[16] D. Gusfield, *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, 1997.

[17] R. Tarjan, “Efficiency of a good but not linear set union algorithm,” *J. ACM*, vol. 22, no. 2, pp. 215–225, 1975.

[18] J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *J. Statistical Software*, vol. 33, no. 1, 2008.