White Paper Report

Report ID: 109365

Application Number: HD-51787-13

Project Director: William Underwood (ted.underwood.3@gmail.com)

Institution: University of Illinois, Urbana-Champaign

Reporting Period: 5/1/2013-11/30/2015

Report Due: 2/29/2016

Date Submitted: 5/1/2015

White Paper
Digital Humanities Start-Up Grant, Award HD5178713

Understanding Genre in a Collection of a Million Volumes

Ted Underwood
University of Illinois, Urbana-Champaign
May 1, 2015

Contents

1 SUMMARY.

One of the main problems confronting distant reading is the scarcity of metadata about genre in large digital collections. Volume-level information is often missing, and volume labels aren't in any case sufficient to guide machine reading, since poems and plays (for instance) are often mixed in a single volume, preceded by a prose introduction and followed by an index. Human readers don't need explicit guidance to separate these parts of a text, but machines do. So the vast resources of digital libraries are still mostly inaccessible for distant reading.

Our goal in this project was to show how literary scholars can use machine learning to select genre-specific collections from digital libraries. We've started by separating five broad categories that interest literary scholars: prose fiction, poetry (narrative and lyric), drama (including verse drama), prose nonfiction, and various forms of paratext. While we have made our code public and published articles describing our methods [27, 25, 29, 28], the deliverables that may most immediately interest other researchers are

- **Page-level maps of genre in 854,476 volumes from HathiTrust Digital Library (English-language monographs between 1700 and 1922, inclusive)** [26]. From this large resource, we have specifically extracted collections of prose fiction, drama, and poetry, where each collection has greater than 97% precision (fewer than 3% of the pages they contain are drawn from other genres).

The genre metadata we have created are available through HathiTrust Research Center. Just as importantly, we have consulted with HTRC to help shape their strategy for the nonconsumptive extraction of features from works under copyright. Since the features they're planning to extract will support our classification workflow, it will be possible to map genres inside works behind the veil of copyright. This may be the place where we need automated mapping most urgently; distant readers would otherwise be flying blind in the twentieth century.

Mapping a broad category like "prose fiction" is of course only a starting point for distant reading. The methods we describe here can also be used to select collections as specific as "detective fiction" or "verse tragedy." However, as we descend from broad categories to subgenres, critical consensus about the list of categories to be mapped becomes less stable. Moreover, while poetry and prose fiction need to be treated for the most part as exclusive sets, the

subgenres within them don't particularly need to be exclusive (e.g. nothing prevents a gothic novel from also being science fiction). This actually simplifies the problem of classification, since categories that are free to overlap can be studied independently. In other words, as we move down the spectrum of sizes, the problem named by "genre" ceases to be a taxonomic question and becomes a largely folksonomic one that does not require a single map or a central controlled vocabulary [30].

2 Project staff; other acknowledgments.

Ted Underwood was PI on this project; he is the author of this report, and he wrote most of the code used for genre classification, drawing on machine learning libraries in Python (scikit-learn) and Java (Weka) [18, 13].

Boris Capitanu is a Senior Research Programmer at the NCSA; he worked with HTRC to design their feature-extraction workflow, and developed web tools for this project as well as a page-level tagger we used to create training data.

Michael L. Black worked on the project first as a graduate research assistant, and then as Associate Director of I-CHASS. He designed the original prototype of the tagger used to create page-level training data, the original version of a Python script we used to extract tabular data from MARC records, and significant parts of other workflows. Shawn Ballard worked on the project as a graduate research assistant, supervising the creation of training data.

Jonathan Cheng, Nicole Moore, Clara Mount, and Lea Potter worked on the project as undergraduate research assistants, playing a vital role especially in the collection of training data.

This is a final report for the National Endowment for the Humanities, which supported the project with a Digital Humanities Start-Up Grant. The PI's time on this project was additionally supported by an ACLS Digital Innovation Fellowship, which was absolutely vital to its success. The project also depended on collaboration and conversation with a wide range of people at HathiTrust Digital Library, HathiTrust Research Center, and the University of Illinois Library, including but not limited to Loretta Auvil, Timothy Cole, Stephen Downie, Colleen Fallaw, Harriett Green, Myung-Ja Han, Jacob Jett, and Jeremy York. Jana Diesner and David Bamman offered useful advice about machine learning.

3 Distant reading and the problem of genre.

The lively debate about distant reading in the last few years may seem to imply that scholars are already studying literary history digitally on a macroscopic scale. In fact, for the most part, we are only beginning to organize the kinds of collections we need for that research. One of the central problems is that scholars don't know how to assemble truly large collections that reliably belong to a particular genre.

Many research teams are creating collections manually, selecting novels or works of poetry one by one, guided by existing bibliographies. Others are borrowing collections from private vendors. Both of those approaches can work—and at the moment, they may be the most practical way for scholars to get started. But they leave an important aspect of the promise of distant reading unfulfilled: the notion that it will allow us to plumb what Margaret Cohen called "the great unread," by taking the library itself as our research collection [7, 17]. Even in the early nineteenth century, there are odd things lurking on library shelves that aren't included in standard bibliographies of literary genres [25]. As we move forward into the twentieth century, we pass increasingly beyond the reach of bibliographies, and it becomes increasingly certain that manually-selected collections will leave out significant swaths of the literary field. Collections based in a public digital library would address at least some of these blind spots; they are also, perhaps more importantly, extensible and interoperable. Instead of building a whole new collection each time we add a genre, period, or problem to our research agenda, we could be defining and redefining collections simply by selecting subsets of the library.

Unfortunately, existing metadata in digital libraries do not provide very strong support for selection by genre. The machine-readable records that accompany volumes do have fields where genre could be indicated—but in practice, genre information is often missing.

Terrible confusion matrix based on received metadata						
# of words	Drama (predicted)	Fiction	Nonfiction prose	Poetry	Paratext	recall
Drama (actual)	2,320,961	79,141	4,690,533	7,852	3,292	32.7%
Fiction	0	1,953,638	3,221,738	0	5,322	37.7%
Nonfiction prose	97,201	26,318	14,534,285	147,780	19,748	98.0%
Poetry	1,650	100,852	1,345,420	678,399	2,940	31.9%
Paratext	9,271	62,224	472,517	36,007	46,745	7.5%
precision	95.5%	87.9%	59.9%	78.0%	59.9%	Microavg F1: 65.4%

Figure 3.1: A confusion matrix for a sample of 370 English-language books 1700-1899, based on volume-level information in MARC records, plus reasonable guesses about "front matter" and "back matter."

In Figure 3.1, for instance, we see how difficult it would be to separate genres if we relied only on volume-level information in MARC records. Volumes that were accessioned before the MARC system was introduced often have very sketchy genre metadata, so if we relied on cataloging information, we would only catch about a third of poetry, fiction, and drama in HathiTrust (see the *recall* column). This is not a hypothetical problem. Scholars actually often search for metadata tags like "fiction," find a very limited selection, and conclude that HathiTrust doesn't have the coverage they need; the present author has seen conclusions of that sort in personal communication.

Even if genre was reported reliably at the volume level, it might not be enough to support distant reading, because volumes are almost always internally heterogenous. Volumes of poetry begin with prose introductions. Late-nineteenth-century novels may end with ten pages of publishers' advertisements and a date due slip. Including those pages in a large corpus can cause strange glitches; one can already glimpse some of the consequences in Google Books' corpus of "English Fiction," which in fact includes many pages of publishers' advertisements in the late nineteenth century. Words common in ads for books, like "cloth," peak in that period.

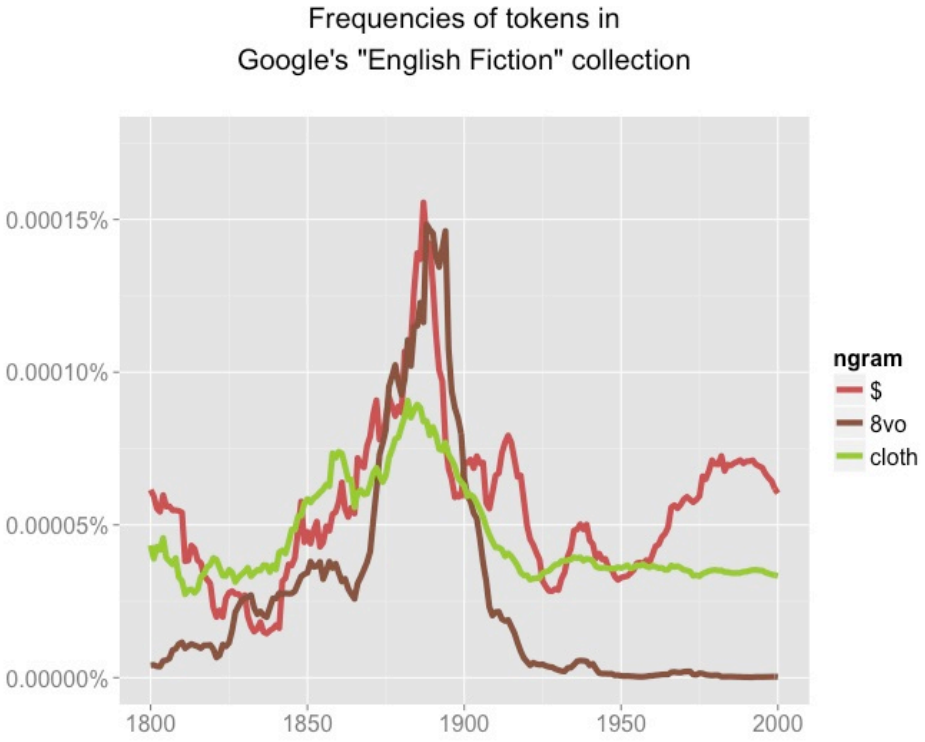

Figure 3.2: Suspiciously volatile word frequencies suggesting that Google's fiction collection actually includes a lot of publishers' ads.

3.1 Goals of this project.

This project proposed to show scholars how to map genre algorithmically in large digital libraries, and to actually generate a page-level map of a million English-language volumes from 1700 to 1949. So far, we have achieved the goal for 854,476 non-serial volumes from 1700 to 1923. But we still plan to extend the map to 1949, and will do so as soon as HathiTrust Digital Library releases information about word counts in volumes beyond 1923; that release is currently delayed while a final layer of concerns about security and intellectual property law are addressed, but we expect it to take place in Spring 2015.

3.2 Deliverables.

Work on this grant has had a number of consequences, including publications [27, 25, 29], and code that is freely available on the web [28]. But the deliverable of most interest to literary scholars will probably be the sets of page-level metadata we have produced [26]. In particular, we have defined **collections of poetry, fiction, and drama, 1700-1923, that are more than 97% precise** (that is, fewer than 3% of the pages in the collection will come from some other genre). Maximizing recall is more challenging, but we can certainly improve the current situation (recall of 30-40%). Depending on the genre, we can get recall of 70-95% while retaining very high precision; scholars who are willing to accept reduced precision can get recall above 87% for all genres.

3.3 What is genre, and can it be mapped at all?

Centuries of literary scholarship have failed to produce human consensus about genre. So scholars are likely to view the notion that genre can be mapped by computers very skeptically indeed.

Several pathbreaking works have shown that digital methods can be surprisingly fruitful here [15, 2]. But if an algorithmic approach to this problem required us first to stabilize an ontology of genre, skepticism would be warranted. Critics don't have an exhaustive list of genres. In fact, we don't have much reason to assume that there is a single coherent phenomenon corresponding to our word "genre" at all. The persistent debate surrounding the term suggests that it may have been stretched to cover a range of different things [20]. "Genre" can be applied to categories that have visible formal characteristics (a lyric poem or an index). But it's also applied to things

that are closer to modes, cutting across formal boundaries (tragedy, romance, and the gothic, for instance, are narrative patterns that recur in different media). "Genre" sometimes describes patterns that persist tacitly for a couple of decades (silver-fork fiction), and sometimes describes institutions that persist for centuries, explicitly organizing bookstores as well as the conscious intentions of authors (science fiction, the novel).

In the 1970s and 1980s, linguists and structuralist critics sought to resolve these problems by giving literary categories a firmer grounding in a deep taxonomy of modes or text types (e.g., narrative / description) [12, 4]. But the project encountered a range of internal contradictions [8], and the prevailing emphasis in contemporary genre theory is nominalistic. Scholars tend to treat genre as a social phenomenon rather than a symptom of deeper linguistic structures. The discipline of rhetoric, for instance, identifies genres with recurring patterns of rhetorical practice; sociologists study specific social differentiations mediated by the categorization of art [9, 11].

Our project has similarly approached genre as an empirical social problem. Moreover, since "genre" may cover a range of social phenomena, with different degrees of blurriness or stability, we've adopted different strategies for different aspects of the problem. About certain broad categories people have, in practice, achieved a great deal of consensus. When we organized a group of five readers to characterize pages by genre, we were able to agree 94.5% of the time about the divisions between, for instance, prose and poetry, fiction and nonfiction, body text and paratext. Even these categories are social phenomena with fuzzy boundaries: the 5.5% of cases where we disagreed were not simply errors. But there was in practice a broad central area of agreement.

Narrower generic categories tend to be less stable, for a variety of reasons. Readers may identify too deeply with specific categories to agree on a controlled vocabulary. You might call this book fantasy, or just science fiction, while I insist on placing it more specifically in a tradition of "weird fiction." Narrow categories can also specialize in different aspects of the composite concept "genre." We might call this book a vampire story, emphasizing subject, or a teen romance, emphasizing audience and plot. And even if we could agree on an ontology or controlled vocabulary, readers would often disagree about the application of tags to individual cases. The group of five readers involved in this project agreed with each other only 76% of the time if we consider their specific claims about narrower categories like "lyric poetry," "epistolary fiction," and "autobiography."

We've concluded that classifying texts by genre becomes a different kind of problem at different levels of specificity. At the broadest level, where "genre" shades into "form," it's possible to define a provisional taxonomy. For instance, "prose fiction," "drama," and "front matter" can be treated for the most part as mutually exclusive categories, and literary scholars have typically found it useful to treat them that way. But as we move into the narrower subgenres within these divisions, genre may more closely resemble a folksonomy, where each text can bear multiple tags, and different readers use different sets of tags [30]. For an interesting example of work on genre that fully exploits the folksonomic dimension of the problem, see Allington [1].

3.4 Why map genre algorithmically?

Algorithmic mapping can be useful for both the taxonomic and folksonomic dimensions of genre. Some of its advantages are obvious. For instance, the problem of scale requires little comment. HathiTrust holds more than 13 million volumes; even the subset of English-language volumes in this project runs to more than 276 million pages. It would take a long time to crowdsource genre classification at that scale.

Other advantages may be less obvious, because it has become folk wisdom that computers can only handle crisp binary logic. If you tell a computer that a novel both is, and is not, an example of detective fiction, the computer is supposed to stammer desperately and emit wisps of smoke.

In reality, the whole point of numbers is to handle questions of degree that don't admit simple yes or no answers. Statistical models are especially well adapted to represent fuzzy boundaries—for instance, by characterizing an instance's mixed membership in multiple classes, or by predicting its likelihood of membership in a single class as a continuous variable.

One important reason to map genre algorithmically is that it allows us to handle these fuzzy boundaries at scale. Human readers can acknowledge gray zones when we're examining individual cases, but in a large crowdsourcing project, the challenge of coordinating different people tends to compel simplification. Our bibliographies and library catalogs are required to decide whether each volume is fiction or nonfiction, and if it is fiction, whether the tag "detective fiction" ought to be applied. Probabilistic models can treat all of these boundaries as questions of degree, by attaching a probability to every prediction. It then becomes possible to sort large collections, not just for clear examples of a category, but for ambiguous ones.

Even a relatively stable boundary like the one separating fiction from nonfiction involves many interesting edge cases that can be highlighted by sorting (we have described a few recently on *The Stone and the Shell* [25]). Mapping ambiguity at scale will be even more important for subtler folksonomic categories. The meaning of a concept like "detective fiction" or "gothic novel" is contained not just in clear examples of the category, but in a wide penumbra of texts that borrow some of its features.

Of course, the ambiguity of generic concepts may also involve kinds of complexity that aren't adequately represented by a single fuzzy boundary. Science fiction can be defined in a range of different ways, which imply *different* boundaries. But this form of ambiguity, too, can potentially be addressed by statistical modeling. Given different sets of examples, it's not difficult to train multiple models of the same genre, and describe how they behave differently.

3.5 Can we actually model genre algorithmically?

The flexible approach to genre I've just outlined may sound appealing in principle, but can we actually *do* any of it? How can we expect a computer to recognize the limits of "detective fiction," or for that matter, the boundary between fiction and nonfiction itself?

3.5.1 In what sense are we "modeling" genres?

When human readers are asked to define the boundaries of genres, we tend to invoke general concepts. When we distinguish fiction from nonfiction, for instance, we tend to think that we're asking ourselves whether the events described in a book really happened. A computer obviously won't be able to answer that. The statistical models of genre deployed in this project are based instead mostly on word counts. Past-tense verbs of speech, first names, and "the," for instance, are disproportionately common in fiction. "Is" and "also" and "mr" (and a few hundred other words) are common in nonfiction. As the Stanford Literary Lab memorably observed, there's something unsettling about the gulf between our abstract concepts of genre and the contingent clues that can be used to recognize them [2].

In what sense can counting words give us a "model" of fiction?

A short answer is that we're creating *predictive* rather than *explanatory* models [22]. An explanatory model attempts to identify the key factors that cause or define a phenomenon. For instance, if we believed fiction could be

defined as a) narrative, b) prose, c) describing imaginary events, we might feel that a properly explanatory model should separate those three factors, and identify appropriate proxies for them.

A predictive model doesn't claim to reproduce this sort of deep structure. As Leo Breiman explains in a now-classic article, it starts by accepting the phenomenon to be modeled as something whose internal logic is "complex and unknown" [5]. Instead of attempting to capture the original causal logic of the phenomenon, it looks for an adequate substitute: a function that maps predictor variables onto response variables in a roughly equivalent way. The validity of the model is assessed simply by measuring its predictive accuracy—in other words, its accuracy on instances outside the sample originally used to train the model.

For humanists, one advantage of these methods is that they don't require structured data. Social scientists can often frame explanatory models, because they start with a relatively small set of clearly relevant variables (age, income, voting record). But humanists almost always begin with an unstructured mass of texts, images, or sounds. In this situation, it's a huge advantage that predictive models don't require researchers to begin by identifying key variables. Instead, you can throw all the evidence you have at an algorithm, and let the algorithm sort relevant from irrelevant features. It's still possible to overfit the model if you provide too much evidence, but out-of-sample predictive accuracy will tell you when that begins to become a problem.

One could also argue that predictive modeling embodies a skeptical epistemology that is deeply appropriate in the humanities. Humanists are often grappling with concepts that aren't yet fully understood, and one of the strengths of the humanities is our ability to acknowledge this. Predictive modeling adopts the same agnostic posture as a starting point, inasmuch as it acknowledges that we are simply observing patterns whose underlying explanation remains "complex and unknown."

In some cases, it may even turn out that predictive models provide more explanatory insight than we would have gathered from an avowedly explanatory model shaped by received ideas. For instance, if you open a random book to a random page and ask yourself whether you're looking at fiction, you may not decide by determining whether the events described "really happened." (We don't usually know the answer, any more than computers do.) When we recognize a book as a novel, we are in practice identifying a particular style of narration. In the process of producing training data for this project, I've

had to make that kind of snap judgment hundreds of times, and I find that I actually rely on much the same clues as the statistical model does—I look for first names and past-tense verbs of speech.

3.5.2 How well do these models actually perform?

Computer scientists tend to ask how well predictive models can reproduce human "ground truth." But this notion of ground truth is a fiction: human beings agree 100% of the time about almost nothing. Algorithms also need to be compared to actual levels of human dissensus. Taking this approach, we've found that statistical models of genre are often only slightly less accurate than human crowdsourcing. I've mentioned that five human readers agreed with each other 94.5% of the time about broad categories like prose fiction, poetry, drama, nonfiction prose, and paratext. Relying in part on voting, we extracted a consensus human opinion from the reports of individual readers. Our statistical model agreed with the consensus only slightly less often than human readers agreed with each other (it achieved 93.6% accuracy overall, fivefold cross-validated). So the short answer is that algorithmic models can predict human opinions about genre almost as well as other humans do.

Moreover, there was a weak but significant correlation between the pages where human readers disagreed and the pages where a statistical model failed. So when we talk about algorithmic "accuracy" with respect to genre, we're talking partly about simple error and partly about a gray zone where human beings and algorithms are forced to make similar definitional choices. A broad definition of fiction might include travel writing with invented dialogue; a narrow definition might contract the category to novels.

The trade-off between precision and recall gives us a way to think about these definitional choices in the context of predictive modeling [6]. A strict definition of a genre maximizes precision (how many of the pages we assigned to genre X were actually in that genre?) A loose definition maximizes recall (how many of the pages that might belong to the genre did we catch?) To some extent these choices are made through the selection of training examples, but there are also ways to let end users tune the model, trading recall against precision to suit their own definitional preferences. In the final datasets produced by this project, we associate a confidence score with each volume so users can set their own precision threshold. This makes it possible to create narrowly-defined collections where 97-98% of the pages are drama or poetry, even though the overall accuracy of our model was 93.6%.

Statistical modeling tends to be less accurate when applied to subgenres like "epistolary fiction" or "detective fiction." (We trained a model of epistolary fiction, for instance; it was only 85% accurate.) But then, human consensus about those categories also tends to be much shakier.

4 METHODS.

Because predictive modeling is unfamiliar in literary studies, we've spent a lot of space above on the philosophical rationale for modeling genre. But what did we actually do, and how did we do it?

4.1 OVERVIEW OF THE PROCESS.

This section provides a brief summary. Details of particular problems are explored in more depth in the subsections that follow.

We began by obtaining full text of all public-domain English-language works in HathiTrust between 1700 and 1922. Organizing a group of five readers, we asked them to label individual pages in a total of 414 books; this produced our training data. We transformed the text of all the books into counts of features on each page; most of these features were words that we counted, but we also recorded other details of page structure. Drawing on the Weka machine-learning library, we wrote software that trained a regularized logistic model for each genre, using a one-versus-all method (each genre was contrasted to all the other genres collectively). Then we tested the models by cross-validating them (in other words, we asked the models to make predictions about the pages of unseen volumes that had been held out from the training set). Because page sequence contains important information (a page sandwiched between two pages of fiction is likely to be fiction), we also trained a hidden Markov model on page sequences in our training set, and used that model to smooth page-level predictions in the test set. We assessed accuracy by looking at the proportions of pages and words accurately classified in each genre.

After trying many different combinations of features and algorithms (see Section 6 for an exhaustive list) we were convinced that regularized logistic regression was the best solution for our purposes, and we settled on a list of 1062 features that maximized predictive accuracy (these included 1036 words and 26 "structural features" that reflect other information about a page or a volume).

We had originally planned to train different models for each century, and transition gradually between them. But in practice we found that it worked better (for the relatively stable categories we were modeling) to sacrifice historical specificity and just use as much training data as we could. For reasons more logistical than philosophical, we did train two slightly different models, one covering the period 1700-1899 and one covering 1725-1922; we used the latter model to make predictions about volumes between 1900 and 1922. Predictions for each volume were saved as a separate JSON file.

Our final concern was to give users a way of adjusting genre definitions to be more or less inclusive by tuning the tradeoff between precision and recall. In principle, we could have done that at the page level, but it seemed more likely that users would want to select texts as volumes, and we also had more predictive leverage at the volume level. We therefore created a new set of models to define volume-level confidence metrics. Since our original page-level models were probabilistic, we could have estimated confidence simply by looking at the original model's degree of certainty about each page, and we did use those degrees of confidence as one piece of evidence. But in practice, it turned out to be more important to consider other kinds of evidence at the volume level. For instance, volumes where our predictions hopped back and forth frequently between genres were often unreliable.

We recorded degrees of confidence for all 854,476 volumes we mapped. A researcher can define their own corpus of fiction, poetry, or drama by setting whatever confidence threshold they desire, and pulling out the volumes that meet that threshold. But we also created special pre-packaged "filtered" collections for these three genres by identifying confidence thresholds for each genre that increased precision for relatively little loss of recall. These filtered collections are available as separate tar.gz files.

4.2 Selection of training data.

This was probably the most questionable aspect of our methodology, and an area we will give more attention as we expand into the twentieth century.

There are two basic problems. One is that labeling individual pages is simply a labor-intensive process. There are ways to abbreviate it a little. We built a page-tagging utility that gave research assistants a GUI to make the process a little smoother (https://github.com/tedunderwood/genre/tree/master/browser).

The second problem was more fundamental: it's that the categories of

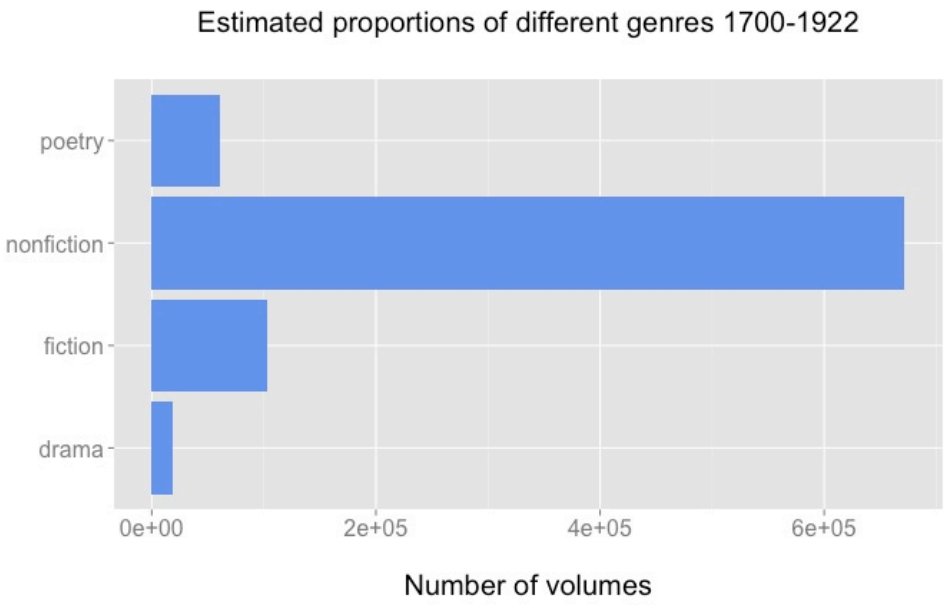

Figure 4.1: Estimated proportions of four genres in English-language volumes 1700-1922.

most interest to our project represented a distinct minority of volumes. How much of a minority, exactly, is something that we didn't know when we began. But having completed a mapping of genres, we can now estimate that nonfiction comprises more than three-quarters of the HathiTrust collection in the period 1700-1922. Poetry, drama, and fiction together make up about 21% of volumes; most of that is fiction. So if we selected training examples completely at random, we would have tagged mostly nonfiction. The imbalanced ratio between genres might not itself have been a problem, but since this is also a labor-intensive process and the total number of volumes we can tag is quite limited, the absolute numbers of volumes tagged in poetry and drama would have been very low. That would have been problematic.

The problem of "imbalanced classes" is a well-recognized one in machine learning, and several papers have shown that it's not necessary for training data to mirror the original proportions of classes [14]. Better results can sometimes be achieved by sampling equal numbers of instances in every class [31]. We accordingly sought to sample roughly equal numbers of volumes in poetry, prose fiction, drama, and prose nonfiction, while also ensuring that our training examples were distributed evenly across time. (The dataset is strongly

skewed toward the end of the timeline, so random sampling across time would have produced very few examples for discriminating genres in earlier periods.)

In general, this was probably a good approach, but there are devils in the details. For instance, how do you select volumes from different genres in the first place? Volume-level metadata is extremely patchy (that's part of the reason why the project was necessary, after all). Selecting from the small number of volumes that happened to have genre tags could introduce unknown biases.

We solved the problem in a couple of different ways, producing training data that is in the end a patchwork of different selection strategies. But our dominant solution—used to select about half of the data—was to use a previous round of volume-level genre classification (using Naive Bayes) as a guide to divide the corpus into genres that we could then sample randomly.

The potential problem here (which occurred to us halfway through the project) is that you might end up with exemplars of genre that are a little better-sorted for purposes of algorithmic classification than a true random sampling of the underlying classes should have been. In other words, your examples of fiction might be drawn disproportionately from examples that were accurately classifiable using word frequencies. It's not a huge problem, because we didn't after all *trust* the genre labels produced by earlier volume-level classification. All our training data was still tagged manually at the page level. And nothing about our sampling strategy would have actually excluded difficult-to-classify volumes. But a small number of difficult-to-classify works of drama would have been dumped into the sparsely-sampled nonfiction pool, whereas the 90% (or so) of drama volumes accurately recognized by Naive Bayes would have been sampled at a higher rate.

To reduce the possible effect of this sampling bias, we later supplemented our training data with works that were selected absolutely randomly from the whole corpus. After doing that, we did notice that classification accuracy dropped by almost a full percentage point—which may suggest that our earlier training data had been unrealistically well-sorted.

Another potential problem with our sampling strategy is that "nonfiction" covers a multitude of problematic things, and undersampling nonfiction might blind you to the risks presented by some of them. For instance, we didn't have many dictionaries in our training data, and dictionaries turn out to be significant ringers.

4.3 CATEGORIES RECORDED.

We explicitly did not set out to create an encompassing taxonomy of genres that could serve as a shared standard. Instead, our research was organized by the specific goal of separating a few broad categories that literary critics do in practice tend to separate: prose fiction, drama, and poetry. We lumped or split other categories as necessary in order to improve predictive accuracy on our literary targets.

For instance, bibliographies and publishers' advertisements are actually very different things. But when reduced to feature sets they look pretty similar (they have lots of capitalized lines, and words like "vol" and "8vo"). We accordingly lumped these categories together, because that turned out to make it easier for our model to recognize both forms of paratext. On the other hand, we separated biographies and autobiographies from other forms of nonfiction, because that separation allowed us to better model the especially challenging boundary between biography and prose fiction. (We borrowed this general strategy from practice at Google, where they similarly capture complex categories by dividing them into parts and training multiple classifiers [21].)

Because we knew that decisions about lumping and splitting would be contingent on predictive accuracy, we didn't attempt to make these decisions while gathering training data. Instead we recorded a relatively long list of detailed categories (anything we thought we *might* need to separate) and then decided later how to group them for purposes of classification.

The boldfaced terms below in the "broad" column are the ones for which we trained classifiers in the present phase of research. All of the categories that follow a broad category are lumped in with it for training purposes. But all those categories are preserved in training data; we might attempt to add finer divisions at a later date.

broad	specific	description
non	non	Nonfiction prose that doesn't belong to the more specific categories below.
	trv	Nonfiction about travel.
	ora	Orations and sermons (that seem to have been actually delivered orally).
	let	Personal letters (actually written as correspondence.)
	pref	Nonfiction located before body text, especially before ToC. Preface, dedication, introduction.

	argum	Prose argument preceding a poem.
	notes	Prose notes that follow a poem or drama.
	errat	Errata slip.
bio	bio	Biography.
	aut	Autobiography.
fic	fic	Fiction.
	epi	Epistolary fiction.
poe	poe	Nondramatic verse.
	lyr	Lyric poetry.
	nar	Narrative poetry.
dra	dra	Drama (could be verse or prose, or mixed).
	vdr	Drama written entirely or almost entirely in verse.
	pdr	Drama written entirely or almost entirely in prose.
	clo	"Closet drama"—poetry in the form of dramatic dialogue that doesn't look like it was ever really intended to be performed.
front	front	Any front matter not otherwise categorized.
	title	Title page.
	toc	Table of contents.
	impri	A page following the title page that lists the authority for publication.
	bookp	A library bookplate, usually in the front of a book.
	subsc	List of subscribers, or any page that's mainly a list of names.
back	back	Back matter not otherwise categorized.
	libra	Text added by the library other than a bookplate—especially, for instance, a due date slip or library information at the back of the volume.
	index	Index. This category also stretched to include alphabetically organized reference material, for instance in dictionaries.
	gloss	Glossary.
ads	ads	A publisher's catalog of titles, or other ads.
	bibli	Bibliography.

There are a number of things that will seem strange about this list, but probably the strangest is the way categories of paratext are grouped into "front

matter," "back matter," and "ads." One could persuasively object that these are structural or formal categories, not genres. And "indexes" are not always located in the back of a book! But in practice accuracy was higher when we lumped indexes with back matter, and predictive accuracy was our primary reason for caring about paratext.

Another, perhaps more importantly debatable boundary is the one that separates "poetry" from "drama." Verse drama is also dramatic poetry, so one could have chosen to treat drama and poetry as overlapping categories. (Some, but not all, pages of drama would also be examples of poetry.) There's no technical obstacle to doing this: multilabel classification, where examples can belong to more than one class at once, is a well-established subfield of machine learning [23]. The problem is literary: in practice, a lot of verse drama drops into prose from time to time. So a division between verse and prose drama would run right through the middle of many pages. And it's not immediately clear what we would accomplish by making that division: in practice, scholars are more likely to organize corpora around the generic concepts of "poetry" and "drama" than around the largely-formal concept of "verse." Moreover, it would be fairly easy to separate verse from prose later, if one needed to do that for some reason.

However, the bottom line is that all the boundaries we drew are debatable. We tended to group short dramatic monologues with lyric and narrative poetry, but one could argue that they belong with drama. The boundary between fiction and nonfiction was also profoundly blurry. How much invented dialogue can history contain before it becomes fiction? How many didactic excursions can fiction contain before it becomes a thinly-disguised treatise on temperance? Questions like this are part of the reason different human readers agreed about only 94.5% of pages.

223 volumes were tagged by five people, with assigned volume lists overlapping so that almost all the pages in the volumes were read by at least two readers (and some by three). This strategy allowed us to make tentative estimates of human dissensus, which were invaluable. But it was a relatively slow process, because it required coordination. The remaining 191 volumes were simply tagged at the page level by the PI. In cases where we had three readers, we resolved human disagreements by voting. In other cases, we accepted the more general genre tag, or the tag produced by more experienced readers.

4.4 Feature engineering.

We used 1062 features in our models. 1036 of them were words, or word categories; a full list is available on Github: https://github.com/tedunderwood/genre/blob/master/data/biggestvocabulary.txt. In general, we selected features by grouping pages into the categories we planned to classify. We took the top 500 words from each category, and then grouped the words from all categories into a master list that we could limit to the top N most frequent words. This ensured that our list contained words like "8vo" and "ibid" that might be uncommon in the whole corpus, but extremely dispositive as clues about a particular class of pages. We normalized everything to lowercase (after counting certain forms of capitalization as "structural features") and truncated final apostrophe-s.

In addition to things that are technically words, our list of features includes several things that are really names for categories, grouping together lots of tokens that, individually, might have been too uncommon to make the list. These included:

- arabic1digit
- arabic2digit
- arabic3digit
- arabic4digit
- arabic5+digit
- personalname
- placename
- propernoun
- romannumeral

The lists of place names and personal names we used are available on Github, under /features/rulesets. Obviously, these are not exhaustive lists, but categorization doesn't have to be exhaustive to improve performance. Personal names are mostly first names, which made them useful for recognizing fiction. When a word was capitalized, not found in a dictionary, and not included in our lists of personal names or place names, we treated it as a generic propernoun.

We also counted words not otherwise included in this feature list and included them as a feature, labeled wordNotInVocab.

The process of counting words also involved a few subtleties where running headers are concerned. The headers at the top of a page often convey especially important generic information—like "Introduction," "Index," or "Act I, Scene II." And we had already, for other purposes, developed an algorithm that could separate recurring headers from body text. So we simply up-voted certain terms if they appeared in a header. "Index" appearing in the body of a page counted once, but "index" appearing in a page header counted three times.

In addition to words or word classes, we included 26 "structural features" that reflect various other kinds of information about a page or a volume. This list metamorphosed quite a lot in the course of the project. We tried a lot of different possibilities here, many of which did not help or actually hurt predictive accuracy. We kept only the ones that seemed to help. For instance, if it seems odd that we count commas, exclamation points, and question marks, but not periods—the answer is simply that periods empirically didn't help.

- "posInVol" = pagenum / totalpages
- "lineLengthRatio" = textlines / mean lines per page in this vol
- "capRatio" = number of capitalized lines on page / number of all lines
- "wordRatio" = words on page / mean words per page in this vol
- "distanceFromMid" = abs(0.5 - posInVol)
- "allCapRatio" = words in all caps / words on this page
- "maxInitalRatio" = largest number of repeated initials / textlines
- "maxPairRatio" = largest number of repeats for alphabetically adjacent initials / textlines
- "wordsPerLine" = total words on page / total lines on page
- "totalWords" = total words on page
- "typeToken" = types on page / tokens on page
- "commasNorm" = commas normalized for wordcount

- “textLinesPerLine” = lines containing any text / all lines
- “typeTokenSqrd” = is literally just typeToken times itself
- “exclamationsNorm” = exclamation points normalized for wordcount
- “questionsNorm” = questions normalized for wordcount
- “endWithPunct” = Proportion of lines ending with punctuation. (The actual formula was (endwpunct + 0.1) / (textlines + 0.3); normalization constants actually mattered here.)
- “endWithNum” = Proportion of lines ending with a digit as either of last two chars. (The actual formula was (endwnumeral + 0.01) / (textlines + 0.2).)
- “startWithName” = Proportion of lines starting with a word that might be a name.
- “startWithRubric” = Proportion of lines starting with a capitalized word that ends w/ a period.
- “capSequence” = Largest number of capitalized initials in alphabetical sequence.
- “capSeqNorm” = Sequential caps normalized for the number of capitalized lines. (The actual formula was (sequentialcaps + 0.2) / (caplines + 2.0). Again, Laplacian normalization constants actually mattered here.)
- “logTypeToken” = type token normalized (multiplied) by the log of sumAllWords. (The formula was typeToken * Math.log(sumAllWords + 50.0).)
- “absWordRatio” = absolute deviation from word mean for vol, normalized by word mean. (Math.abs(sumAllWords - meanWordsPerPage) / meanWordsPerPage.)
- “metaBiography” = a flag based on metadata telling us this is biography
- “metaFiction” = a flag based on metadata telling us that this is fiction

Some things here have an obvious purpose (features that indicate the position of a page in a volume help separate front and back matter from body text). Others may be a little more opaque. We put a lot of effort into

designing features that could separate indexes from body text. This is hard in the nineteenth century, because indexes don't have the kind of stable format they will later acquire. Some of the features above are designed to identify pages where a lot of lines begin with the same letter (maxInitialRatio) or where lines that begin with capital letters tend to be arranged in alphabetical order (capSeqNorm).

Laplacian normalization can be very important for some of these features. Otherwise blank pages, or pages with very few words, end up having extreme values, which can make the feature useless or actually harmful. In cases where we added small constants to denominator and numerator, that's the reason.

Metadata features deserve special comment. It seemed reasonable to suppose that existing volume-level metadata would be a useful "prior" guiding page-level classification. We tried to take advantage of genre tags, titles, and Library of Congress call numbers. But in practice most of these seemed not to help. I'm honestly a little perplexed as to the explanation. One part of the reason may be that genre information for poetry and drama was simply too sparse. And Library of Congress call numbers don't reliably indicate genre; they indicate subject. Still, you might have thought that a broad division between call numbers in the Ps and others would provide at least a regulative hint about the likelihood that a volume is nonfiction. Perhaps we would need a more sophisticated (hierarchical) model to appropriately combine volume-level and page-level clues.

The one exception to this rule involved genre metadata about biographies (including autobiographies) and fiction. Although these tags were not consistently available in volume-level metadata, they were present often enough to provide useful assistance with this tricky boundary.

4.5 Algorithms and code.

We wrote the core of the classification workflow in Java. This choice was originally shaped by concerns about execution speed and concurrency, but it also turned out to be fortunate that we were using a language that compelled object-oriented discipline, because the project became more sprawling than we had envisioned.

For most actual machine-learning algorithms, we used the Weka implementation [13]. But our code is very far from being a mere wrapper for Weka, because the grouping of pages in volumes gave this classification problem a peculiar structure.

Many of the features that characterize individual pages are *relative* to counts calculated at the volume level. For instance, we calculate the number of words on a page, relative to the average number of words-per-page in this volume. So we needed Volume objects that organize (and in fact create) page instances. Also, when we're cross-validating a model, it's vital that we don't allow pages from a single volume to appear at once in the training set and the test set. Otherwise we might be learning to recognize a particular set of volumes, with no proof that the model is able to make reliable predictions about previously unseen volumes.

Finally, after probabilistic page-level predictions have been made, there's a smoothing step, where a hidden Markov model uses observations about the probabilities of transition between different genres to smooth the genre probabilities assigned to individual pages. This, too, has to be cross-validated; we don't want to use observations about a volume to smooth page sequences in the same volume.

Once we had designed this overall workflow, it was possible to plug different classification algorithms into the page-level classification step of the process. We tried a range of algorithms here, including random forests and support vector machines. We also tried a range of different ensemble strategies, including strategies that combine multiple algorithms, before settling on an ensemble of regularized logistic models, trained by comparing each genre to all the other genres collectively.

Our preference for regularized logistic regression (aka ridge regression, aka MaxEnt modeling) is not necessarily based on proof that it's absolutely the optimal algorithm for this dataset. It's possible, or likely, that exhaustive tuning of support vector machines could eventually produce slightly better results. But datasets of 100,000 pages, with 1000+ features, were large enough that the training time for ensembles of SVMs created significant delay. We were also dealing with a workflow where lots of other things kept changing: new data, new approaches to feature engineering, and so on. Using an algorithm that could be trained relatively quickly made it possible to optimize other aspects of the workflow.

4.6 VOLUME STRUCTURE.

The strictly page-level part of this project was, from a machine-learning perspective, pretty routine. We extracted features, we applied regularized logistic regression, we predicted class probabilities for each page.

The only part of this that might be novel or interesting for computer scientists involves the grouping of pages in volumes. There are intellectual challenges involved in coordinating page-level information with information extracted from page sequence. (E.g., indexes are pretty likely to follow nonfiction, and not very likely to precede fiction. Pages of drama are likely to occur next to other pages of drama.)

There are a variety of clever approaches that might be tried to coordinate page-level predictions with knowledge of volume structure. We trained a hidden Markov model, which is is a relatively simple approach. The model contains information about the probability of transition from one genre to another, so it is in a sense a model of volume structure. But in practice, its main effect is to smooth out noisy single-page errors—for instance, it was good at catching a few isolated pages misclassified as nonfiction in the middle of a novel. We added a bit more sophistication with an ad-hoc rule that relaxed the model slightly whenever it encountered a blank page (or actually, a page with fewer than five words). This reflected the reality that blank or nearly-blank pages often represent divisions between volume parts, so the genre probabilities on either side of such a page should propagate only weakly across the boundary. This was a relatively minor change, but it did improve performance.

More sophisticated approaches to volume structure, using conditional random fields, or a maximum-entropy Markov model, ought to improve results by reflecting more kinds of volume-level information. We did experiment with other approaches here, but weren't able to find a solution that actually improved on the HMM reliably, in a cross-validated test. However, the success of our volume-level confidence metrics shows that there is more useful information to be extracted at the volume level, if one could find the right way to apply it back to page-level inference.

4.7 Divisions below the page level.

There are of course many cases where genres mix on a single page. Volume sections are usually separated by pagebreaks, but sometimes an introduction gives way to body text, or body text gives way to an index, right in the middle of a page. Lyric poems are often inserted in other texts as quotations or epigraphs. Editions of Shakespeare may share the page with editorial footnotes. The running headers at the tops of many pages are technically paratext.

Our decision to divide volumes at the page level isn't meant to imply that all these other divisions are unimportant. Rather, it reflects a judgment

that we're looking at different kinds of problems here, and there's not much to be gained by tackling them all at once. The statistical models applied here wouldn't work very well to disentangle short passages of poetry or drama from prose. To do that, you'd want to focus much more on sequences of capitalized lines, and also on line length and white space; possibly the principled way to do it is to train some kind of Markov model at the line level. That will be worth doing, but there's no reason why it has to be done at this stage; it makes sense to envision it as a later stage of processing, focused on page ranges where we've identified a probability that drama or poetry are present.

Other page-level issues (running headers, for instance) may require different methods, and some of the issues are not going to affect enough text to pose a real problem for distant reading. Running headers can matter, because the words involved get repeated many times. But if a four-hundred-page book gives way to an index midway through the final page, we can probably live with the noise. Distant reading is not scholarly editing.

5 Results.

We trained two different models, one that we used for the period 1700-1899, and one that we used for volumes dated 1900-1922. In the pages that follow we present confusion matrices for both models. But actually, the differences between them are relatively minor.

Influenced by a blog post written by Michael Witmore, the original proposal for this grant considered a range of different ways to allow definitions of genre to change over time [33, 27].

But in the end we didn't actually find historical specificity very useful for the genres we've emphasized in this report. When we attempted to train models that focused on a particular century, we always found that they were less accurate than models covering a two-century timeline. There are at least three possible explanations for this.

- Broad genres like "poetry" are not actually very volatile in the period covered here.
- Since HathiTrust volumes are dated by year of publication rather than year of composition, every year contains reprints from a wide range of earlier years, and you need a wide range of training data to handle this diversity.

- Historical specificity might become useful with larger amounts of training data. But given the limited amount of data we had, it was simply never a good idea to cut it in half.

We don't have evidence yet that would allow us to decide between those explanations. In the end we did train two different models covering slightly different spans of time, but this decision was shaped mainly by the difficulty of managing a complex workflow and terabytes of data. (We hadn't absorbed the 1900-1922 data yet at the time we ran the first model.)

Confusion matrix for first model, 1700-1899 (counting words)

	Drama (predicted)	Fiction	Nonfiction prose	Poetry	Paratext	recall
Drama (actual)	6,773,616	61,797	90,981	57,877	3,019	96.9%
Fiction	3,056	5,516,684	542,972	14,235	1,079	90.8%
Nonfiction prose	206,718	254,092	17,923,395	130,652	62,334	96.5%
Poetry	34,072	14,751	127,514	2,174,359	1,313	92.4%
Paratext	20,227	5,516	138,032	35,823	587,856	74.7%
precision	96.2%	94.3%	95.2%	89.7%	90.1%	Microavg F1: 93.9%

Figure 5.1: Confusion matrix for the first model, five-fold cross validation.

5.1 Evaluating confusion matrices.

Confusion matrices are presented here in the form we actually used to tune the models. The numbers in each cell are numbers of words. For instance, in cross-validation this model misclassified 3,019 words that were actually drama as paratext. It correctly classified 6,773,616 words of drama. These word counts are generated simply by assigning all the words on each page to the genre associated with the page; we haven't actually attempted divisions below

the page level. But it made sense to count words in the tuning process, because errors about blank or nearly-blank pages are less important than errors about text.

The microaveraged F1 measure reported in the yellow cell is, in effect, simply accuracy: the overall number of pages assigned to the correct genre.

Although we trained separate models for "biography" and "nonfiction," we treated both models as a vote for nonfiction when estimating accuracy. We weren't after all trying to distinguish biographies from other forms of nonfiction, we were just training multiple models as a way of capturing the fiction/nonfiction boundary more precisely. The same logic governed front matter, back matter, and advertisements. We trained three separate models, but in practice we counted all three of them as a vote for the broader category of "paratext."

Overall accuracy declined very slightly in the second model, trained on data from 1725 to 1922. It's difficult to say whether there is actually much significance to this; it could easily be random noise in the training data.

Confusion matrix for second model, 1725-1922 (counting words)

	Drama (predicted)	Fiction	Nonfiction prose	Poetry	Paratext	recall
Drama (actual)	6,645,782	52,896	40,438	57,594	2,492	97.7%
Fiction	2,202	5,248,642	833,759	23,820	972	85.9%
Nonfiction prose	291,743	255,420	19,760,554	110,504	71,886	96.4%
Poetry	33,354	10,7277	112,139	2,173,973	1,819	93.2%
Paratext	15,896	4,969	149,396	32,711	724,745	78.1%
precision	95.1%	94.2%	94.6%	90.6%	90.4%	Microavg F1: 93.2%

Figure 5.2: Confusion matrix for the second model, five-fold cross validation.

Paratext was consistently the hardest problem. Advertisements and indexes can be difficult to distinguish from nonfiction, and title pages may

sometimes look like poetry. We can claim to have significantly reduced the amount of paratext mixed into literary genres, but not to have eliminated it altogether.

It may also be worth comparing these results to the initial results we reported in Summer 2013 [27]. At that point, with a training set of 100 volumes, microaveraged F1 was 89.4%. It took about six months of work to get that up to 93.6% (averaging both models). We did it partly by quadrupling the amount of training data, and partly by fine-tuning feature selection and feature engineering. I think that was worth doing, but I don't think it would be a good idea to spend a great deal more time optimizing this phase of the process. During the last month of labor, we tried a lot of clever ideas, but progress was in reality fairly stationary.

One advantage of regularized logistic regression is that it's relatively interpretable. A model with 1062 features depends on a lot of different factors, and it's somewhat misleading to extract a few that happen to be at the top or bottom of a list. Sometimes a single feature becomes important in effect as a counterweight against lots of others. But it may be interesting to note features that were unexpectedly important in particular genres.

The features most strongly predictive of fiction were structural features (typeTokenSqrd, typeToken, totalWords, lineLengthRatio). Also "the," "my," "you," "said," and "her." The features that argued most strongly against fiction were Arabic numbers.

The features most strongly predictive of poetry were, again, structural (typeToken, capRatio, wordRatio, totalWords, and endWithPunct). A wide range of words that might be categorized as poetic diction were also predictive ("doth," "songs," "sweet," etc.) Features arguing most strongly against poetry were structural (exclamationsNorm, wordsPerLine, typeTokenSqrd, logTypeToken). It's interesting that typeToken appears on opposite sides of this model, depending on the way it's transformed or normalized.

Features strongly predictive of drama were stage directions ("enter," "exit," "exeunt"), terms like "scene" and "act," as well as a different set of structural features (startWithRubric, wordRatio, and startWithName). A motley set of features, hard to generalize, argue against drama, but past-tense verbs of speech like "exclaimed" and "said" stand out; generally, past-tense verbs of speech characterize fiction.

Regularization was relatively light in these models; a ridge parameter of around .002 maximized accuracy.

5.2 Distribution of genres across the timeline.

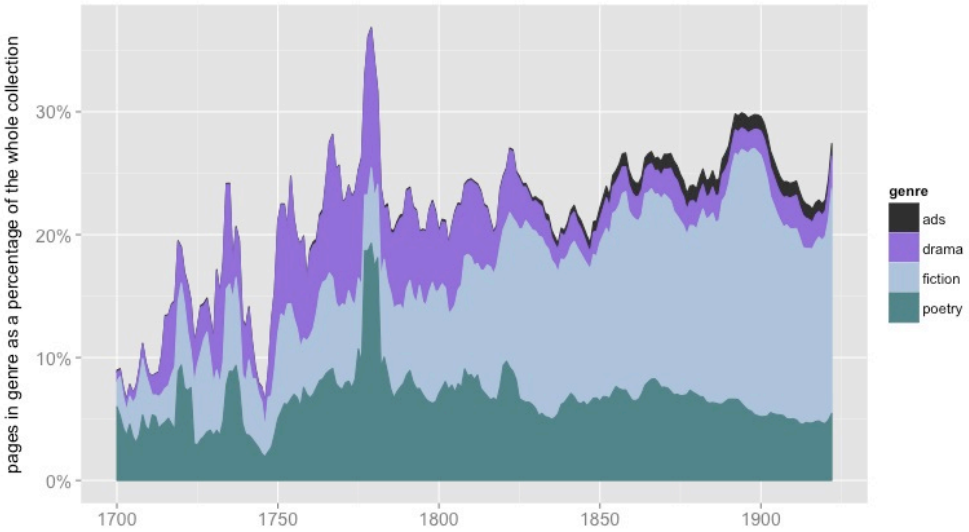

Figure 5.3: Pages in HathiTrust, five-year moving average.

We've counted the pages assigned to each genre in each year. Nonfiction dominates so much that it's better to leave it out to make other changes visible.

This very simple analysis isn't likely to reveal much about literary history that we don't already know, but there is still perhaps some interest in seeing familiar trends visualized. The volatile fluctuations in the eighteenth century are mostly a consequence of small sample sizes, although the deep dip right before 1750 might have something to do with the Licensing Act of 1737, at least where drama is concerned. The nineteenth-century decline of drama, and corresponding rise in prose fiction, is more or less what received literary history would lead us to expect. The late-nineteenth-century increase in pages devoted to ads is less familiar, and affects a smaller number of pages, but it could produce distortion if you were counting, say, dollar signs. (Technically, the category "ads" can also include some pages that are bibliographies, but in practice these are mostly publishers' advertisements.)

Of course, what we're really visualizing here are the digitized portions of library collections. This is not a picture of everything that was printed, and is in fact fairly certain to underrepresent some categories of printing, like unbound pamphlets and popular "story papers." Scholars of book history will also want evidence that comes more directly from publishers.

5.3 TUNING RESULTS TO SUIT THE NEEDS OF INDIVIDUAL USERS.

As indicated in Figure 3.1, volume-level metadata had originally been so spotty that recall for nineteenth-century fiction, poetry, and drama would be less than 40% if you relied on it. It's not at all hard for algorithmic methods to improve on that level of recall! The confusion matrices in figures 5.1 and 5.2 show that page-level classification raised recall to 86-96% while keeping precision reasonably high (it ranged from 90% for poetry to 94-96% for drama and fiction).

These are pretty good results overall, especially if you care equally about precision and recall. The potential problem is that literary scholars may *not* care equally about those factors. When we focus on a genre, we're used to working with very small collections that implicitly exclude most of what was published (they have terrible recall), but that include almost no false positives (they have very high precision). Whether these habits are good or bad, they are likely to shape disciplinary norms. It seems likely, for instance, that many scholars will be unwilling to work with a collection of poetry that contains up to 10% noise, even if it improves recall.

So it seemed important to give scholars a way of raising precision. One obvious way to do this was to filter results at the volume level. This made sense especially because we knew that there was volume-level information not fully exploited in our page-level model. For instance, pages assigned to a particular genre were especially likely to be errors when they occurred in a volume where that genre comprised only a small portion of pages overall. We had tried, without success, to use this sort of information to improve page-level predictions. But we knew it could be used somehow.

Our solution was to create confidence metrics by training new models on the sets of predictions created by our page-level classifier. A wide range of modeling strategies were explored; the most effective was to train a logistic model in order to predict *the probability that a volume's accuracy would be greater than a fixed threshold T.* (Attempting to model accuracy as a linear function was less effective, probably because it's not a linear function.) We trained models to predict both the overall accuracy of genres assigned to pages and the levels of precision in particular genres (poetry, fiction, and drama). For instance, a volume with only five pages of fiction may have high accuracy overall, but since those five pages are likely to be errors, its predictions about fiction are likely to have low precision.

The predictive features used were:

- A flag that indicates whether volume-level metadata coincides with the majority of page-level predictions. In the case of genre-specific metrics, this flag reported specifically whether volume-level metadata coincided with that genre.

- Another metadata flag indicating whether volume-level metadata specifically controverted page-level predictions. We needed two flags because there's also the (very common) case where metadata tells you nothing at all.

- The percentage of pages in the most common genre (for the overall accuracy metric). In genre-specific precision metrics, this feature was supplemented by another reporting the percentage of pages in the genre of interest.

- The percentage of times genre predictions "flipped" in the raw predictions (before smoothing by HMM). Proceeding through pages sequentially, a "flip" is a change of genre as we move from page n-1 to page n. Volumes with small groups of pages flipping back and forth frequently between genres were likely to contain lots of errors. This was reported as a percentage (normalized by the total number of pages).

- The absolute number of times genre predictions "flip" in the final predictions (after smoothing by HMM). This number was not normalized.

- The average gap between most-likely-genre and second-most-likely genre in the probabilistic predictions made by page-level models.

- The average probability of the most-likely-genre in the probabilistic predictions made by page-level models.

We trained these models of confidence on our full collection of training data (414 volumes). Then we examined the subsets of training data that would be produced by using various settings of the confidence metrics for particular genres as a threshold to filter out volumes. This allowed us to plot a relationship between precision and recall in particular genres. Because 414 volumes is after all not a huge set, the resulting curves had a jagged granularity caused by single incidents of exclusion or inclusion; we smoothed them conservatively with a loess function. The curve for fiction is plotted in Figure 5.4.

Each dot here indicates the precision and recall to be expected in a dataset limited to volumes at or above a given confidence threshold. The large red dot

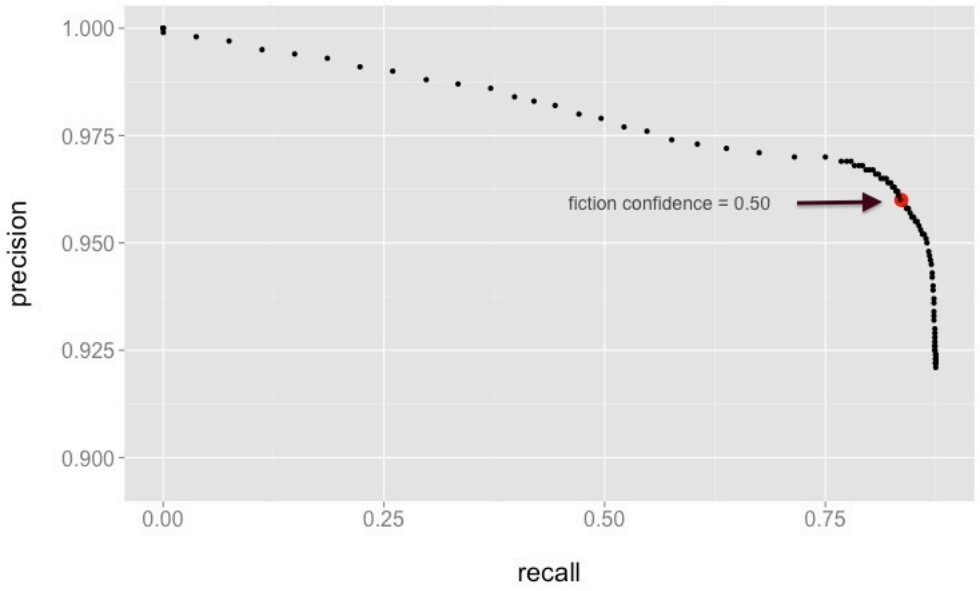

Figure 5.4: Precision-recall graph for fiction datasets limited to various thresholds of confidence. Graphics from ggplot2 [32].

indicates the threshold we actually chose to produce a filtered fiction dataset. Our model assigned a confidence threshold of 0.50 to a volume if it expected that particular volume's predictions about fiction to have a 50% chance of being more than 80% precise. But practice, if you construct a dataset with *all* volumes at this threshold *or higher*, you get 96.0% precision with 83.5% recall (since most of the volumes included in that dataset will, after all, have a confidence metric much higher than 0.50).

The dramatic "elbow" in this curve suggests that there's a lot to be gained by filtering at the volume level. As we start to raise the confidence threshold, we initially get a lot of gain in precision with very little loss of recall. The volumes we're discarding, in other words, mostly contained pages mistakenly predicted to be fiction. After rising through the elbow, the situation reverses: we start to lose recall rapidly with minimal gains in precision. At this point we're uselessly discarding actual fiction. A threshold located on the elbow arguably gives you the best deal.

In the case of poetry, the situation isn't quite as pretty, because the elbow is located at a lower point in the graph, and we were concerned that it might not provide enough precision for researchers (see Figure 5.5). We accordingly pushed the confidence threshold for poetry in our pre-packaged dataset a

little higher, to a point beyond the elbow in the graph. However, the key thing to realize is that we've incorporated these curves in the JSON metadata for all volumes where we predicted any pages of poetry, drama, or fiction. So if a researcher is concerned, say, that our filtered dataset excludes too much poetry, they can relatively easily recreate their own filtered dataset with a confidence threshold designed to create the precision-recall tradeoff they feel most appropriate.

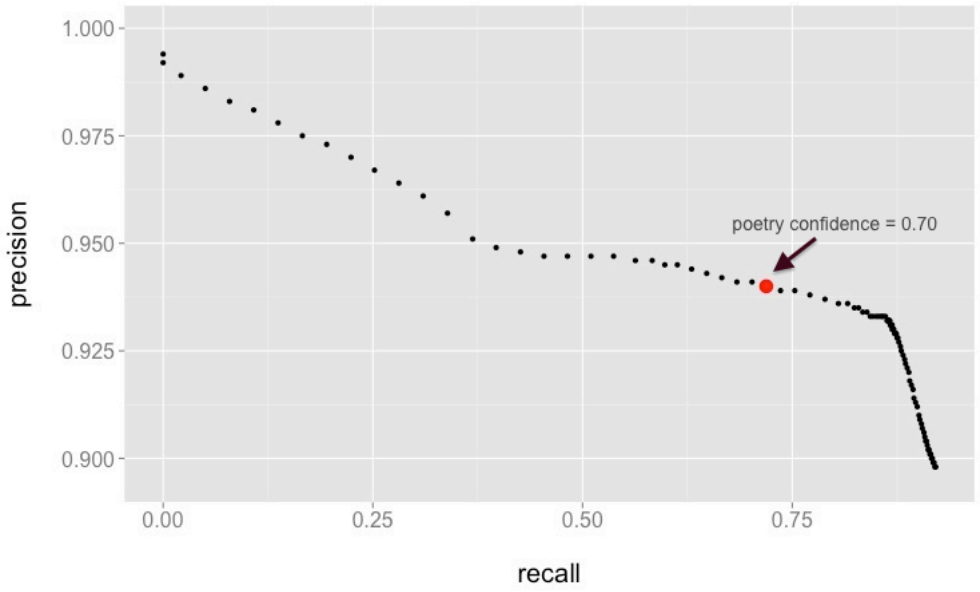

Figure 5.5: Precision-recall graph for poetry datasets limited to various thresholds of confidence.

5.4 Pre-packaged datasets.

To make it easier for researchers to profit from volume-level filtering, we have created pre-packaged datasets of poetry, drama, and fiction, using relatively high confidence thresholds. Looking at these datasets, we also realized that manual filtering could improve things a bit further. It would be impossible to manually tag genres in 276 million pages, but once you have a pretty-good list of 100,000 fiction volumes, it's actually not hard to find a few hundred obvious ringers. One of the strategies we used was to sort the volumes by size and scan the titles of the largest volumes. This was effective both because oddly large volumes tended to be things like encyclopedias, dictionaries, Congressional

reports, etc. that didn't belong in the list, and because it meant we were likely to catch *big* errors—a relatively effective use of our limited time and energy.

Since predictions about precision based on a training set of 414 volumes are not necessarily to be trusted very far in a collection of 854,476 volumes, we also confirmed precision by randomly sampling 300 pages from each filtered dataset. We stratified each sample to get 100 pages from each century, because we also wanted to ensure that precision wouldn't vary dramatically across time (10% noise that varies across time can, actually, be worse than 15% that doesn't). In Table 5.4, we report predicted precision, predicted recall, and observed precision ranges across all centuries of the dataset.

Genre	Threshold	Pred. precision	Pred. recall	Observed precision
drama	0.66	98.9	94.8	97-100
fiction	0.50	96.0	83.5	96-99
poetry	0.70	94.1	70.2	97-99

Predicted precision and recall are based on a model that was assessing numbers of *words* correctly classified; the observed figures are based on observations of *pages*. But the real reason precision is generally higher than predicted is because of manual filtering. (Without that filtering it would probably have been slightly lower than predicted; researchers who want to create their own filtered datasets are advised to also exclude volumes from the list of "ringers" we have provided.) The "observed precision" column is reporting the range of precisions observed in all three centuries. For fiction, eighteenth-century precision was lowest, but for drama the low point was the nineteenth century, and for poetry it was the twentieth. Since the eighteenth century is quite a small part of the dataset, you may get a better sense of overall precision by considering only the period 1800-1922. For drama, precision in this period would be 98.5%; for fiction it would be 97.5%; for poetry it would be 98%.

5.5 How to use this data.

We believe these datasets represent a new and valuable resource for distant readers. Even after filtering to increase precision, these are large collections (18,111 vols of drama, 102,349 vols of fiction, 61,286 vols of poetry). Once HathiTrust Research Center has completed page-level feature extraction, it will be possible for anyone to download page-level feature counts for the whole collection, and then use our genre predictions to create large datasets representing particular genres.

5.5.1 Strengths and weaknesses of coverage.

For good or ill, these datasets are also significantly different from the kinds of collections researchers tend to construct from bibliographies or acquire from vendors. Those datasets are often keyed to dates of first publication, and contain only one record for each work. HathiTrust, by contrast, is an aggregation of public and university libraries; popular works are often represented by multiple editions and reprintings scattered across time. The collection also includes a lot of works in translation, anthologies, and works for juvenile readers. All of these forms of inclusion could be an advantage or a disadvantage, depending on your research goal. Perhaps most importantly, the size of the collection increases the range of alternatives we have available.

No single collection of works is universally representative, because there are a range of different things a literary historian might want to represent. What were authors writing in 1871? What was getting printed or reprinted? What were people of a given class (or gender, or race, or age) reading? What was getting reviewed? What got bought by libraries? These are all valid questions. In its raw form, HathiTrust probably comes closest to answering the last of them. But HathiTrust is also a very large collection, which means that researchers have a lot of scope to select subsets and rebalance the collection as they see fit. For instance, it's possible to select a smaller collection that focuses tightly on a particular genre, geographic region, or demographic group. Even if you end up selecting a few thousand volumes from HathiTrust mostly by hand, it might save time to rely on our predictions for the page-level mapping inside volumes.

A few particular limitations of HathiTrust's coverage are worth remark. Although the datasets we have produced cover the period from 1700-1922, HathiTrust's coverage of the early eighteenth century is somewhat spotty, and other resources might well be preferable in that period.

Up to about 1820, it's necessary to be aware of the dangers posed by OCR on old typefaces, especially the notorious "long S." This is not by any means an insuperable obstacle, just something to be aware of. We corrected OCR before running our classification workflow, and may eventually distribute word counts for those corrected texts [24].

Special caution is also necessary where drama is concerned. In fiction and poetry, reprints affect the overall trajectory of the corpus only to a moderate extent; we've compared this collection to smaller collections of first editions, and the differences are not particularly shocking. The case of drama may be different, because Shakespeare, and other Elizabethan dramatists, make up a large proportion of the drama bought by universities in this period.

5.5.2 Data format and concrete workflows.

If you're working with HathiTrust Research Center, you could use this data to define a workset in their portal, by specifying a list of volume IDs. Alternatively, if your research question can be answered with word frequencies, you could download public page-level features from HTRC and align them with our genre predictions on your own machine to produce a dataset of word counts associated with a particular genre. You can also align our predictions directly with HathiTrust zip files, if you have those.

The **pagealigner** module in our Github repo is intended as a handy shortcut for people who use Python; it will work both with HathiTrust zip files and with HTRC feature files.

For each volume, we produce a set of page predictions saved as a separate JSON object in utf-8 encoding. The filename convention is HathiTrust volume ID + ".json"; for instance, "coo.31924013573666.json." Those individual volume files are aggregated in .tar.gz files that cover a whole period, or a particular genre. If you're using the **pagealigner** module, you can leave the .tar.gz files compressed; the script reads directly from the tarfile. For more information, consult documentation accompanying pagealigner.py in the repository, under /utilities [28].

Data for each volume is represented as a separate JSON object in a file whose name is the HathiTrust volume ID + ".json."

All files will contain the following top-level keys:

added_metadata: This is an object containing keys for *genre_counts, maxgenre* and *totalpages.*

genre_counts, in turn, is an object with keys for all the genres in this volume, pointing to the number of pages in each genre. Note that the counts here may not add up to the value of *totalpages,* because pages tagged "bio" are also included in the count for nonfiction. So if you simply total all the genre_counts, pages of biography will be counted twice. Get the total number of pages from *totalpages.*


```
{
  "added_metadata": {
    "genre_counts": {
      "back": 7,
      "bio": 2,
      "dra": 108,
      "front": 17,
      "non": 8,
      "poe": 104
    },
    "maxgenre": "dra",
    "totalpages": 244
  },
  "hathi_metadata": {
    "author": "Wordsworth, William,",
    "genre_tags": "",
    "htid": "coo.31924013573666",
    "inferred_date": 1919,
    "title": "The complete poetical works
              of William Wordsworth"
  },
  "page_genres": {
    "0": "front",
    "1": "front",
...
    "14": "front",
    "15": "front",
    "16": "front",
    "17": "poe",
    "18": "poe",
    "19": "poe",
    "20": "poe",
    "21": "poe",
...
```


hathi_metadata: This is not meant to substitute for the full metadata available through HathiTrust; it's mainly a pointer to allow you to retrieve that metadata, plus a few fields to give you a sense of what you're looking at. It always contains keys for *author, genre_tags, htid, inferred_date* and *title.* The *htid* is the HathiTrust volume ID. Inferred date will be the earliest of several dates that may be provided for the item in the MARC record. *Genre_tags* here are extracted from the HathiTrust MARC record, not produced by our mapping, and users are advised to take them with a grain of salt.

page_genres: Contains numbers for all pages from 0 to totalpages-1, pointing to the most probable genre for each page. Possible values are "non," "bio," "fic," "poe," "dra," "front," "back," and "ads." Numbers will always be sequential, and their sequence maps onto the sequence of individually-numbered page files in HathiTrust (although the numbers in the two sequences are not guaranteed to be the same). Note that none of these numbers map onto the page numbers printed on the original text. The **pagealigner** module in our repo under /utilities may simplify some of these tasks if you use Python.

This volume demonstrates some of the advantages of page-level genre classification. It's a volume of *Wordsworth's Poetical Works,* but the bulk of the volume is actually occupied by Wordsworth's tragedy *The Borderers*. Our model has correctly assigned the first part of the volume to poetry, and the latter half to drama. It also captures a brief nonfiction preface separating *The Borderers* from the poetry that preceded it.


```
...
   "113": "poe",
   "114": "poe",
   "115": "non",
   "116": "non",
   "117": "dra",
   "118": "dra",
   "119": "dra",
   "120": "dra",
...
   "230": "dra",
   "231": "dra",
   "232": "poe",
   "233": "poe",
   "234": "poe",
   "235": "poe",
   "236": "poe",
   "237": "back",
   "238": "back",
   "239": "back",
   "240": "back",
   "241": "back",
   "242": "back",
   "243": "back"
 },
 "volume_accuracy": {
   "precision@prob": 0.973,
   "prob>95acc": 0.53,
   "recall@prob": 0.768
 }
 "drama": {
   "dra_precision@prob": 0.993,
   "dra_recall@prob": 0.942,
   "pages_dra": 108,
   "pct_dra": 0.44,
   "prob_dra>80precise": 0.72
 },

 "poetry": {
   "pages_poe": 104,
   "pct_poe": 0.43,
   "poe_precision@prob": 0.947,
   "poe_recall@prob": 0.537,
   "prob_poe>80precise": 0.79
 },
}
```


All volumes will also contain a key for **volume_accuracy:** This in turn contains a confidence metric, *prob>95acc*: the probability that we have correctly identified genres for more than 95% of pages in the volume. The other two keys here are not reporting the (unknown) precision and recall for this volume, but estimating the precision and recall that would be produced if you constructed a corpus of all volumes at this confidence threshold *or higher.*

The last two keys here (**drama** and **poetry**) are only contained in JSONs for volumes where some pages have been assigned to those genres. If any pages had been assigned to fiction, a parallel object for fiction would be included. Each object lists the number and percentage of pages assigned to the genre, as well as a confidence metric calculated specifically for that genre. For instance, *prob_poe>80precise* reports the probability that more than 80% of pages assigned to poetry were correctly so assigned. *poe_recall@prob* reports the recall that would be produced by including only volumes at this confidence threshold or higher.

5.5.3 METADATA FORMAT.

For full metadata records associated with volumes, we refer researchers to the range of HathiTrust bibliographic services (http://www.hathitrust.org/data). The .csv files we have provided (ficmeta.csv, and so on) are not intended as a substitute; a lot of information has been abbreviated or left out. They're provided purely for the convenience of researchers who may want to get a quick overview of what's included in the collection covered by our metadata before deciding whether to invest more time in it.

Brief explanations of the columns in the metadata files:

htid: HathiTrust volume ID. This is an item-level identifier for volumes. IDs used for pairtree storage can come in two different forms; we have provided the "clean" versions of the IDs that are legal as filenames [16].

recordid: This is the HathiTrust record ID; in multivolume works, it may be the same for all volumes.

oclc: OCLC number, when available.

locnum: Library of Congress call number, when available.

datetype, startdate, enddate, imprintdate: The first three of these fields are extracted from MARC controlfield 008 (http://www.loc.gov/marc/archive/2000/concise/ecbd008s.html). The last is extracted from other text fields in MARC.

place: Place of publication code, as documented in MARC controlfield 008.

enumcron: A field that distinguishes multiple volumes of a single record.

prob80precise: In genre-specific datasets, this column indicates the probability that more than 80% of the pages assigned to the genre in this volume are correctly so assigned.

genrepages, totalpages: In genre-specific datasets, the number of pages in the relevant genre, and the total number of pages in the volume.

6 LESSONS LEARNED.

Researchers are frequently advised to record failures and blind alleys as well as successes. Taking that advice to heart, the PI of this project kept a running list. We did encounter a lot of blind alleys, and a record of them might be useful for literary scholars attempting any large data mining project.

6.1 THINGS THAT WERE ABSOLUTELY WORTH THE EFFORT.

- *Building a GUI to streamline production of training data.* This took a great deal of work by Michael L. Black and Boris Capitanu. But it was worth the early investment of time. In particular, it was useful for the GUI to permit "rapid page-flipping" in situations where you know the next 400 pages are going to have the same genre.

- *Parallelizing software.* Multi-core hardware is only useful if your software takes advantage of it. Weka doesn't automatically parallelize logistic regression, so Underwood wrote Java code to parallelize at a higher level, training models for different genres concurrently. This mattered, because you do a lot of trial and error, and it matters whether each trial takes two hours or four.

- *Training volume-level models of confidence on top of the underlying model.* These occurred to us only at a late stage of the process, but they were valuable. After banging one's head against a wall for months trying to optimize the overall accuracy of the page-level model, it was a relief to discover how much easier it was to improve precision by trading away small amounts of recall.

6.2 THINGS THAT DID NOT HELP.

- *Hadoop.* We spent part of a summer on this, and for our problem it wasn't worth the effort. If you had >5TB of data, it might start to be necessary. But that's pretty uncommon for humanistic problems. With 2TB of data, you're better off parallelizing across lots of flat files.

- *Cutting-edge algorithms.* We tried support vector machines and random forests as well as logistic regression. SVMs are usually the gold standard for text classification, and it's probably true that an exhaustive grid search would eventually reveal settings where the SVM outperforms our regularized logistic model. However, a quick grid search didn't reveal those settings. And in a complex workflow with a lot of moving parts, we didn't feel it was worthwhile to rely on algorithms that might require a week of tuning every time some other aspect of the workflow changed.

- *Complex ensembles.* There's something deeply attractive about ensemble learning, and we had really hoped to produce a solution that would

involve an ensemble of different algorithms, with different feature sets. We spent the better part of a month combining random forests with logistic regression in various ways. However, in the end it turned out that logistic regression with a relatively large feature set was the best solution; the various forms of boosting and bagging we tried didn't actually improve performance.

- *Sophisticated approaches to the multiclass problem.* Most classification algorithms are designed to make a binary prediction; if you want to choose between more than two possible classes, you have to find a way of organizing binary predictors to "vote" about a multiple-choice question [3]. For instance, you could train one classifier for each class, contrasting it to the examples of all other classes, or train one classifier for each class *boundary* (fiction-against-poetry, fiction-against-front-matter, fiction-against-drama, and so on). The first strategy is sometimes called one-vs-all; the second, all-versus-all. There are also further refinements [10]. The secondary literature is rather confusing here; a number of articles acknowledge that one-versus-all is commonly used, but then propose that some other more refined solution is better [3]. No doubt that's sometimes true. However, for our use case we found that one-vs-all worked best, and there is some theoretical support for the notion that it's a robust solution [19].

- *A bunch of other stuff specific to this task.* Active learning. Co-training. Using volume-level metadata as priors. Using call numbers. Creating a special category for travel literature. Separating autobiography from biography. Training classifiers for particular centuries. Organizing classifiers in a cascade structure (may have helped a little, but effect was small and not worth the added complexity).

6.3 Things that still need doing.

This is a final report for the funded stage of the project, but work is still underway to make the resources we built more easily accessible to a community of scholars without technical expertise. I've already released the metadata I generated [26], and HathiTrust Research Center has just released public data for 4.8 million volumes in the public domain (https://portal.htrc.illinois.edu/features). So researchers can take my list of volumes and

pages in specific genres and match them up with HathiTrust's data to create datasets today.

IP law prevents HTRC from releasing the texts themselves, but they have been able to release extracted *features,* for instance wordcounts, that in practice can support a lot of significant digital research. At the moment, researchers would need to align my metadata with HathiTrust's data, but I can save them a step by using my page-level metadata to actually gather in one location the public, extracted data about all volumes, and pages, predicted to contain English-language fiction, poetry, or drama, and by making those datasets easily downloadable for researchers. That's something I plan to do by June 2015.

Sometime in 2015, HathiTrust Research Center will also start releasing features from works published after 1923. And crucially, I've communicated to them which features I would need to support mapping of fiction, drama, and poetry in that period. So I'll be able to use their extracted features to produce page-level maps of genres in books after 1923. This may take longer, because I'll need to generate more training data; I would anticipate releasing further datasets in 2016.

The datasets I've generated have already supported a number of scholarly articles, as well as a public-humanities intervention in *Slate* [27, 25, 29]. They will continue to support scholarly interventions both in my research and more widely through the NovelTM project (`http://novel-tm.ca`). The genre-specific datasets I create are likely to be disseminated both through NovelTM and through HathiTrust Research Center.

I don't anticipate extending the project back before 1700; EEBO is probably a more appropriate resource in that period.

6.4 How portable is this code?

We've made our code public, and believe parts of it may be portable, but as a whole it definitely is not a tool that could simply be pointed at collections from other libraries, languages, or periods to map them. Our methods can be *adapted* to other problems. But the process of mapping genres involves too many domain-specific aspects to be packaged as a "tool."

For instance, a lot of the labor involved goes into tagging and organizing training data. That's a manual task, and a domain-specific one: you can't export training data very meaningfully across languages or periods. It would be risky even to apply a model trained on one dataset to data that had undergone a different sort of OCR correction. Moreover, it's likely that the optimal set of

features will vary from one period to another; regularization parameters will need to be tuned; decisions about the precision-recall tradeoff will require human judgment.

In short, using machine learning to map genre is at best a months-long project with a lot of domain-specific data munging. If we set out to write code that encapsulated the fully generalizable, portable part of the project ... we would probably end up with something like the library of machine learning algorithms already contained in Weka and scikit-learn [13, 18].

On the other hand, we do hope that the code in our Github repository will provide at least a useful template for other scholars contemplating the same kind of undertaking [28]. Some parts of the code (for instance, the GUI we used to browse and tag volumes at the page level) might conceivably be borrowed unchanged by other research projects. Others could be used more loosely as models. And the repo includes a small library of Python utilities that might be useful for particular data-munging tasks researchers are likely to encounter: for instance, extracting a sequence of pages from a HathiTrust zip file or extracting tabular metadata from a collection of MARCxml.

The aspect of this project that we would more confidently expect to be portable is our general account of what's involved in using machine learning to construct collections for literary research—including methods that work well, and blind alleys that don't. That's why we've made this report so verbose.

REFERENCES

[1] Daniel Allington. Exploring genre on SoundCloud, part I. http://www.open.ac.uk/blogs/vem/2014/06/exploring-genre-on-soundcloud-part-i/, April 2014.

[2] Sarah Allison, Ryan Heuser, Matthew Jockers, Franco Moretti, and Michael Witmore. *Quantitative Formalism*. http://litlab.stanford.edu/LiteraryLabPamphlet1.pdf. Stanford Literary Lab, 2011.

[3] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–41, 2000.

[4] Douglas Biber. A typology of English texts. *Linguistics*, 27(1):389–421, 1989.

[5] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

[6] M. K. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American Society for Information Science*, 45(1):12–19, 1994.

[7] Margaret Cohen. *The Sentimental Education of the Novel*. Princeton University Press, 1999.

[8] Jacques Derrida. The law of genre. *Critical Inquiry*, 7(1):55–81, 1980.

[9] Amy J. Devitt. *Writing Genres*. SIU Press, 2008.

[10] Thomas Dieterrich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[11] Paul DiMaggio. Classification in art. *American Sociological Review*, 52:440–55, 1987.

[12] Gérard Genette. Genres, 'types', modes. *Poétique*, 32:389–421, 1977.

[13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[14] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 Conference on Artificial Intelligence*, pages 111–17, 2000.

[15] Matthew L. Jockers. *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, 2013.

[16] J. Kunze, M. Haye, E. Hetzner, M. Reyes, and C. Snavely. Pairtree for object storage. http://tools.ietf.org/html/draft-kunze-pairtree-01, November 2008.

[17] Franco Moretti. Conjectures on world literature. *New Left Review*, 1, Jan/Feb 2000.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 2004.

[20] Marina Santini. State of the art on automatic genre identification. Technical Report ITRI-04-03, Information Technology Research Institute, http://www.itri.brighton.ac.uk/techindex.html#sthash.U9gy55r3.dpuf, 2004.

[21] D. Sculley, Matthew Eric Otey, Michael Pohl, Bridget Spitznagel, John Hainsworth, and Yunkai Zhou. Detecting adversarial advertisements in the wild. In *KDD '11*. http://www.eecs.tufts.edu/~dsculley/papers/adversarial-ads.pdf, 2011.

[22] Galit Shmuéli. To explain or to predict? *Statistical Science*, 25(3):289–310, 2010.

[23] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[24] Ted Underwood. A half-decent OCR normalizer for English texts after 1700. http://tedunderwood.com/2013/12/10/a-half-decent-ocr-normalizer-for-english-texts-after-1700/, October 2013.

[25] Ted Underwood. Distant reading and the blurry edges of genre. http://tedunderwood.com/2014/10/22/distant-reading-and-the-blurry-edges-of-genre/, October 2014.

[26] Ted Underwood. Page-level genre metadata for English-language volumes in HathiTrust, 1700-1922. http://dx.doi.org/10.6084/m9.figshare.1279201, December 2014.

[27] Ted Underwood, Loretta Auvil, Boris Capitanu, and Michael L. Black. Mapping mutable genres in structurally complex volumes. In *Big Data, 2013 IEEE International Conference On*, pages 95–103. http://arxiv.org/abs/1309.3323, 2013.

[28] Ted Underwood, Michael L. Black, and Boris Capitanu. Code for Understanding Genre in a Collection of a Million Volumes. https://github.com/tedunderwood/genre.

[29] Ted Underwood, Hoyt Long, and Richard Jean So. Cents and sensibility. http://www.slate.com/articles/business/moneybox/2014/12/thomas_piketty_on_literature_balzac_austen_fitzgerald_show_arc_of_money.html, December 2014.

[30] Thomas Vanderwal. Folksonomy. http://vanderwal.net/folksonomy.html, February 2007.

[31] Gary M. Weiss and Foster Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315–54, 2003.

[32] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. http://had.co.nz/ggplot2/book. Springer New York, 2009.

[33] Michael Witmore. The time problem: Rigid classifiers, classifier postmarks. http://winedarksea.org/?p=1507, April 2012.