9-23-2019

# Text Mining and Subject Analysis for Fiction; or, Using Classification, Keyword Extraction, and Named Entity Recognition to Assign Subject Headings to Dime Novels

Matthew Short



Original Citation
This is an Accepted Manuscript of an article published by Taylor & Francis in Cataloging & Classification Quarterly on 23/09/2019, available online: http://www.tandfonline.com/10.1080/01639374.2019.1653413.

# Text Mining and Subject Analysis for Fiction; or, Using Classification, Keyword Extraction, and Named Entity Recognition to Assign Subject Headings to Dime Novels

Matthew Short

*University Libraries, Northern Illinois University, DeKalb, IL, USA*

mshort@niu.edu

Matthew Short is the Digital Collections & Metadata Librarian at Northern Illinois University, where he maintains a collection of more than 6,000 digitized dime novels on the site *Nickels and Dimes* (https://dimenovels.lib.niu.edu). He is Principal Investigator on the Albert Johannsen Project to digitize the dime novels of Beadle & Adams and has written on story papers, sentiment analysis of 19th century popular fiction, and linked data.

# Text Mining and Subject Analysis for Fiction; or, Using Classification, Keyword Extraction, and Named Entity Recognition to Assign Subject Headings to Dime Novels

This article describes multiple experiments in text mining at Northern Illinois University that were undertaken to improve the efficiency and accuracy of cataloging. It focuses narrowly on subject analysis of dime novels, a format of inexpensive fiction that was popular in the United States between 1860 and 1915. NIU holds more than 55,000 dime novels in its collections, which it is in the process of comprehensively digitizing. Classification, keyword extraction, and Named Entity Recognition are discussed as means of assigning subject headings to improve their discoverability by researchers and to increase the productivity of digitization workflows.

Keywords: subject analysis; text mining; cataloging for digital resources; cataloguing popular fiction; dime novels

## Introduction

Text mining uses patterns from unstructured text to derive new knowledge about that text. Although it has been widely studied for the purpose of information retrieval, the topic is not often examined from the perspective of the library cataloger. This paper explores the use of text mining methods to facilitate and improve cataloging efficiency and accuracy, focusing narrowly on subject analysis of fiction. It describes several experiments undertaken over the last five years while working on *Nickels and Dimes*,[1] a collection of more than 7,000 digitized dime novels, and includes an overview of classification, keyword extraction, and Named Entity Recognition. While text mining cannot fully automate subject analysis, it may offer new ways for catalogers to save time and improve the quality of their cataloging.

**Subject Analysis and Fiction**

Assigning subject headings can often be one of the most challenging aspects of cataloging fiction. There are seldom table of contents or introductions, which makes it much more difficult to determine what the book is about, and most of the guidelines and literature about subject access have been primarily written for cataloging non-fiction. Dust jackets, prefaces, publisher statements, author interviews, and reviews can be helpful, but often may be lacking, and as much as they might like to read the entire novel, the cataloger usually only has time to read a few pages or skim for relevant keywords. The subject of some novels can be more elusive than others, particularly if the cataloger goes beyond genre and tries to identify geographic settings, characters, and topics.

But subject analysis can also be one of the most important parts of the description, especially when describing very large collections of fiction about which little is known. This is often the case when cataloging popular fiction from the 19th century, like dime novels, as many as 50,000 of which were published between 1860 and 1915. Although some researchers may know exactly what title or author they are looking for, many are more interested in finding all the stories about Salt Lake City, or telegraph operators, or Chinese Americans, or crossdressing. With so many novels to choose from, identifying every story that addresses a particular subject can be a daunting task, with the potential to discourage scholars who might be unfamiliar with the format. Identifying appropriate subjects and genres for each title can often play an important role in facilitating new research.

While many libraries have had large collections of dime novels for decades, it has only been within the last 20 years that any concerted effort has been made to comprehensively catalog them at the monographic-level with full subject analysis. This

is because, historically, dime novels were cataloged as serials, if they were cataloged at all. Several libraries and archives have recently undertaken massive retrospective cataloging projects, including Northern Illinois University, where cataloging has coincided with large-scale digitization. These projects have provided an opportunity to re-examine existing cataloging with a much more clearly-defined purpose, i.e. building a digital collection, and to consider ways in which to improve cataloging efficiencies to keep pace with the rate of digitization. Text is always extracted from each digitized novel after scanning by means of Optical Character Recognition (OCR), providing large quantities of text and presenting an opportunity to explore how text mining might be used to address both concerns.

**Cataloging Dime Novels**

The dime novel is a format of inexpensive fiction that was popular in the United States between 1860 and 1915. Usually available for as little as a nickel or as much as a dime, the format made leisure reading widely available to an increasingly literate population, variously referred to by contemporary critics and publishers by epithets like the "Unknown Public," the "great people," or the "million." Many genres of popular literature, like the Western,[2] the detective novel, [3] and science fiction,[4] have their origins in the dime novel format, where generic conventions still common today were first developed. Although the quality of the prose is often poor, with formulaic stories consisting mostly of dialogue and often containing problematic depictions of women and people of color, the dime novel featured prominently in the everyday life of many Americans for more than half a century. As such, they can offer unique insights into what a wide range of people were thinking and feeling during the later half of the 19th century, including children, the working classes, immigrants, and women. They also provide fertile ground for exploring the evolution of early genre fiction.

With a few exceptions, most dime novels are unknown to modern readers. Authors like Metta Victoria Fuller Victor and Edward S. Ellis, once household names, are today only familiar to a handful of collectors and scholars. And while there is some scholarship about the dime novel, there is not as much as might be expected when other formats of popular fiction that proliferated in their wake, like comics, are much more widely taught and studied. This is primarily a problem of access. Because so few dime novels are still in print, they can be challenging to research without access to one of only a handful of special collections that specialize in 19th century popular fiction. By making large collections of dime novels available online, scholars no longer need to be geographically adjacent to one of these collections to do their research. NIU has been digitizing dime novels for several years, making thousands of them freely available through the site *Nickels and Dimes*, often for the first time anywhere in more than a century.

However, the problem of access is more fundamental than the access to freely read and view dime novels online. As one researcher recently wrote: "Even with such access, it remains difficult for scholars to know where to start when faced with thousands of ... volumes, all deliberately branded to look as similar as possible."[5] A truly incredible number of novels were written for popular presses in the 19th century, issued in hundreds of series, some of which contain thousands of issues. Much of the scholarship that exists often tends to focus on the same few novels, sometimes making broad generalizations about the entire format based on a small sample. So, while it is true that digitization means that dime novels can be made more widely available, inadequate cataloging means that, in many ways, the novels themselves are still inaccessible.

Most dime novels do not take long to digitize, sometimes consisting of as few as 16 (very tightly-packed) pages. Cataloging can often take much longer, particularly if trying to identify geographic locations, characters, and topics. This means that an entire series can be scanned in the time it takes to catalog only a small fraction of its issues. Catalogers need to be aware of project timelines and requirements, which can often factor into how much time they can spend on subject analysis, typically the most time-consuming part of describing these materials. It was while grappling with the competing demands between efficiency and access that experimentation with text mining in cataloging began.

**What is Text Mining?**

Text is one of the most common forms of data, available as e-books, academic articles, news, tweets, email, online reviews, and much more. While comprehensible to humans, it is unlike other data in that it is largely unstructured and difficult for machines to process. Text mining is a way to analyze collections of unstructured texts in order to reveal hidden relationships between those texts and to uncover themes that might not be immediately obvious through close reading, especially in cases where close reading is not possible because of the number of texts involved. It is being used by digital humanists studying the structure and style of literature and historical trends; by businesses to develop tools for recommending products based on similarity; and by scientists to extract interactions between genes and proteins, such as protein or gene-diseases relationships.[6]

The traditional knowledge discovery framework contains multiple iterative steps, including preprocessing, feature extraction, modeling, and evaluation. Text mining can be considered a single step in that process (i.e. modeling or selecting and applying an algorithm or method) or as synonymous with the entire knowledge

discovery process.[7] This paper prefers the latter definition, since its focus is on how text mining can be used to facilitate the work of the cataloger, rather than the algorithms and methods involved. These algorithms are mentioned but are largely out of scope. Although a background in statistics and computer science certainly helps with this work, detailed knowledge of statistics and programming is not usually required to work with most of the tools mentioned below.

**Preprocessing and Feature Selection**

Raw text is likely to contain many words that will not be relevant to the question being asked, from the OCR noise mentioned above to words that appear frequently, such as definite or indefinite articles. Preprocessing allows us to clean unstructured text, which often improves the quality of models and results. While it can be one of the most tedious and time-consuming parts of any text mining project, it is often also one of the most important. Even those algorithms that take raw text as input will process or prepare the text in some way, even if that preprocessing might be hidden to the operator. The most common preprocessing tasks, briefly described below, are tokenization, filtering, lemmatization, and stemming.

Tokenization involves taking a sequence of characters and breaking them up into tokens--typically words or phrases--by using common boundaries, like punctuation. Punctuation is usually discarded during this step, unless considered important for later analysis. Filtering may be done next to remove stop-words, or words that often appear in a text, but may have little semantic value. This includes prepositions, articles, and conjunctions, like “a,” “an,” “the,” “but,” and so on. A stop-word list might also contain other words that appear frequently or words that appear infrequently, depending on the task. For the classification tasks discussed in this article, Matthew Jocker’s expanded stop-word list for 19th century British, American, and Irish fiction was often used,

which includes thousands of personal names common in the 19th century.[8] The next step is often to stem the remaining tokens, which reduces inflected words to their base or root form. For example, “cataloging” and “cataloged” might be reduced to the morphological root “catalog.” Stemming algorithms are always language dependent, like the Porter Stemmer for English, which was used for all of the experiments described below.[9] Finally, lemmatization involves further morphological analysis, grouping like words together, such as mapping verb forms to their infinite tense. Unlike stemming, lemmatization requires first identifying the word’s part of speech, as well as the meaning of the word within a sentence or the document, making it a more complex and prone to error. Lemmatization is closely related to stemming, although stemming is usually preferred because it is much more efficient.[10]

After preprocessing, the next step is to convert the text to structured data, which involves selecting the features of the text that we are most interested in examining. One common model for feature selection and text representation is bag of words, which considers the number of times a word occurs, while ignoring grammar and word order. This approach makes it possible to represent words as vectors, which can then be analyzed with algorithms from machine learning and statistics. In some cases, especially when dealing with many texts of varying lengths, it is also necessary to go beyond word frequency and rank words based on their significance. One of the most widely implemented methods for ranking words, used in as many as 70% of text-based recommender systems,[11] is TF-IDF, or term frequency-inverse document frequency.

TF-IDF is expressed as term frequency multiplied by inverse document frequency, where term frequency is calculated based on how often the word appears in the text, expressed as a ratio of word occurrence over total number of words, while inverse document frequency is represented as the ratio of documents that contain the

word. The value of TF-IDF increases based on the number of times a word appears in a document, offset by how frequently the word appears in the corpus. This accounts for the fact that some words appear more frequently than others but is intended to adjust for how important a word is in a document relative to its importance in all documents being analyzed.

After a text has been preprocessed and features have been selected, the next step is to design a model. There are a variety of different text mining methods, but two broad categories are the focus of this paper: supervised methods, such as classification, and information extraction, including keyword extraction and Named Entity Recognition. Unsupervised methods, like clustering and topic modeling, are also briefly mentioned at the end. The remaining sections provide an overview of these methods and how they were used to improve cataloging of the dime novel collections at NIU.

## *Nickels and Dimes* and the Texts

Rare Books and Special Collections at NIU is home to two of the largest dime novel collections in the country: the Albert Johannsen and Edward T. LeBlanc collections. Johannsen and LeBlanc were collectors and bibliographers, who devoted much of their lives to describing dime novels. It is no exaggeration to say that the field of dime novel research would not exist today without their dedication and tireless efforts in preserving and promoting the format. Although no detailed inventory has ever been made, the combined collections are estimated at approximately 55,000 volumes, making NIU's holdings comparable in size to the other two major collections of dime novels in the United States, located at the Library of Congress and University of Minnesota. NIU began digitizing its dime novel collections in 2013, making them freely available to read and download on *Nickels and Dimes.* Scanning has gone hand-in-hand with cataloging and the creation of linked data through the Edward T. LeBlanc Memorial

Bibliography, which is hosted by Villanova University.[12] This work relies on existing bibliographies to unpack the complex relationships that exist between editions of the same story and to describe often complicated authorial identities, including thousands of pseudonyms and house names.[13] In 2017, NIU and Villanova were awarded a Digitizing Hidden Special Collections grant from the Council on Library and Information Resources to digitize the entirety of the Johannsen Collection, which has significantly increased the rate of digitization. There are over 7,000 dime novels available on the site, with another 1,000 volumes to be added over the next year.

The text from *Nickels and Dimes* it is extracted from the novels by means of Optical Character Recognition (OCR), a method for electronically converting printed text into machine-encoded text. Because this is an automated process, the extracted text often contains various inaccuracies. The number of errors varies significantly from book to book, depending on factors such as typography, layout, and the condition of the item. Many of the dime novels we are digitizing are in especially poor condition, both because they were typically printed on cheap and acidic paper and because so little effort was made to preserve them. Type also tends to be very small in order to fit as many words as possible onto a single page, and it was very common for the same plates to be in use for several decades, with later impressions sometimes only just barely legible. As a result, some of the OCR that we have extracted is full of "noise": garbage characters created by OCR errors, which mean very little or nothing.

Several dime novels that have been digitized have also been hand-keyed for Project Gutenberg by Villanova University and the Distributed Proofreaders, which makes comparison between the OCR and the clean text of the same novel possible. One popular method for measuring the similarity between texts involves calculating the cosine of the angle between using their TF-IDF vectors.[14] The smaller the angle

between two documents, the larger the cosine value and the greater the similarity. When comparing the TF-IDF vectors from extracted OCR against the TF-IDF vectors from the Project Gutenberg edition of the same novel using scikit-learn in Python,[15] the cosine similarity of both texts is usually somewhere between 70-75%. In most cases, this has proven to be an acceptable accuracy threshold for the activities described below, but results would certainly be improved if using corrected text.

The more challenging aspect of the novels we are working with is how the text is structured. Almost every dime novel includes some front matter and back matter, consisting chiefly of advertisements for the publisher's other dime novels. These will usually be long lists of titles, containing many distinct words. Because the same products are often advertised by the same publisher throughout their line of novels, some words and phrases often rise to the top that signify little about the novels themselves. Many of the dime novels published in the early 20th century also contain significant back matter after the featured novel, in order to make the dime novel appear more like a periodical and so qualify for the lower Second Class postage rate.[16] This might include short stories, serialized stories, jokes, fashion advice columns, or letters from correspondents. Such sections make up a proportionality smaller part of the text for any given dime novel, but they are often the leading cause of any unusual results during text mining. Just as we did not correct the OCR, additional contents were not removed. In the end, we are trying to develop a way to improve cataloging efficiency, which is not possible when resources are needed to manually correct the text.

**Classification**

Classification involves training a model to predict the most appropriate label, or class, to assign to an unseen document. Once trained, the model is used in a text mining application to organize a group of documents into a set of predetermined categories. It is

said to be a supervised learning task because the problem is one of inference from known classes using a training set of labeled documents, with a test set reserved for evaluating performance. The test set is classified using the model, then the predictions are compared against the known labels, with the number of correctly classified documents referred to as the model's accuracy. On the surface, classification seems like an ideal tool for subject analysis, because we have a set of known classes in the form of Library of Congress Subject Headings, with thousands of books that have already been assigned at least one label in their existing records.

Initial experiments to build a classifier for the dime novels were limited to genre headings only, and then to a mere 5 headings from the more than 256 available. These included "Adventure stories," "Western stories," "Love stories" (now "Romance fiction"), "Detective and mystery stories," and "Historical fiction," which are the most common dime novel genres. The first model was created in 2013, using Waikato Environment for Knowledge Analysis, otherwise known as Weka.[17] This is a suite of open source machine learning software tools developed at the University of Waikato, which features both a command line tool and a graphical user interface. The first model for this project was designed to perform a "hard" classification task, explicitly assigning a single genre to each novel.

Because this was very early in the digitization project, there were only 177 novels available with which to train and test the model. Pre-processing involved tokenizing on common word boundaries, converting to lowercase, stemming, and removing stop-words and punctuation, after which words were then converted to vectors using TF-IDF. Several classification algorithms were tested, but k-nearest neighbor performed the best, with an overall accuracy of 82%. Unfortunately, the success of this experiment was almost certainly because the training set and test set

were so small, which overdetermined the model. In other words, the model was very good at identifying genre among that set of novels, which were mostly homogeneous and from the same series, but performed less well when dealing with texts that contained more variables. After digitization had progressed, subsequent attempts to classify 1,900 dime novels (approximately 90,000 pages) using the same model scored less than 60% accuracy, which was less encouraging and suggested relatively few useful applications for cataloging. Even with only five classes, it was obvious that a great many more instances were required before a classification model could be trained that would have any useful application. This was one of the first lesson learned about classification: a significant number of texts would be needed to build an accurate model with several classes.

[Insert Table 1 near here]

[Insert Table 2 near here]

In Spring 2016, a group of students from NIU's Digital Convergence Lab worked with the author to develop a classification model and genre prediction application capable of assigning multiple labels with their predictive probability. Unlike the prior experiment, this was to be a "soft" classification task, intended to assign more than one label to a text (i.e. identify novels that participate in multiple genres). The training and test set, with an 66/34 split, consisted of 1,387 texts, excluding dime novels that contained multiple works and those that participated in multiple genres. With a larger corpus to work with, two additional classes were added to the original five genres described above: "Bildungsromans" and "Sea stories." The former consisted chiefly of stories modelled after the rags-to-riches stories of Horatio Alger, which we usually describe with the local genre term "Success stories." Preprocessing involved converting words into TF-IDF vectors with the Snowball stemmer and Matthew Jocker's stop-word

list, keeping 500 words per class. The classifer application was developed with Weka's Application Program Interface in Java, using the Naive Bayes classification algorithm. Based on the test set of 472 novels set aside to evaluate the model, the initial predictive accuracy score was over 77%. An additional 214 novels, unseen by the classifier, were then independently evaluated by a graduate student from the English department and compared against the model's predictions, resulting in an accuracy of 71%.

Table 1 shows the top ten words determined by the model to be predictive of each genre, which suggests in a naïve way how the model made its predictions. Some of these associations are obvious: words like "sailor" and "schooner" often indicated a sea story, while "criminal" and "disguise" suggested a detective story. But other words are less clear. Most of the stories that had been assigned to the genre "Bildungsromans," for example, take place on Wall Street, which explains why "broker," "market," and "stock" are at the top of the list. (This is another example of how a model can be overdetermined by the texts used to train and test, since many other types of stories fall under the umbrella of "Bildungsromans.") Note also that this list contains the stem "illustr" for the word "illustrator," since the illustrator is attributed in many of the novels used in this set. This is clearly not a word indicative of any genre and should have been removed during the preprocessing step. Table 2 shows the confusion matrix Weka produces, with the predicted class in the columns and the actual class (as assigned by catalogers) in each row.

Although a predictive accuracy of >70% is respectable, the classifier that was ultimately created has limited utility, since assigning genre terms is likely one of the most straightforward tasks when cataloging dime novels. Many series are dedicated to a particular genre, like stories about detectives or cowboys, or else the genre can be easily

discerned from the cover illustration and title. However, the classifier has been useful in at least two unexpected ways.

First, it has proved helpful when evaluating the quality of existing cataloging, especially when it comes to consistency. Several catalogers have worked on NIU's dime novel collections over the last twenty years and there has not always been consensus among them. One major area of disagreement has been in how particular genres are defined. For instance, many of the earliest dime novels are frontier stories set in 16th or 17th century New England, often written in imitation of James Fenimore Cooper's *Leatherstocking Tales*. While these do not take place in the Western United States, the conventions of these stories are very similar to the conventions of the Western genre, which evolved directly from the frontier story. Indeed, the literature on dime novels frequently lumps these stories together as "Frontier and Western stories," since time and place are not always seen as the Western's defining feature. While our current best practice is to label these "Western stories," prior catalogers had instead assigned them to "Adventure stories." As a result, novels in the test set that had been labelled as "Adventure stories" were frequently misidentified as "Western stories" by the application, accounting for many of the classification errors observed. Once revealed, this allowed us to revisit prior cataloging decisions and to codify genre definitions in our local documentation.

The classifier not only revealed inconsistencies in how "Western stories" was being applied, it also pointed to the fact that "Adventure stories" was often being used when some narrower term would be more appropriate. One of the most common classification errors was between "Adventure stories" and "Love stories." Examining these novels revealed one likely explanation: these novels really do not belong to one genre. Instead, they have aspects of "Love stories," in that there is almost always some

romance and a marriage, and they have aspects of "Adventure stories," especially in terms of pacing and the presence of danger. In many ways, "Adventure stories" and "Love stories" are the default dime novel genres in that every novel involves some adventure and some romance. These classification errors revealed that our catalogers had been assigning these two terms any time a more obvious genre term was not unavailable.

Whether catalogers used "Adventure stories" or "Love stories" often had to do with whether the main character was a woman. To be fair, dime novels with female protagonists are often more love story than adventure story, but certainly not always. Below is a plot summary from bibliographer Albert Johannsen of one of these misclassified instances,[18] a book assigned the genre "Love stories" by a cataloger that was misclassified as "Adventure stories" by the classification model:

> The adventures of a young woman in the early part of the nineteenth century. From her home in St. Louis, then headquarters for the fur trade, she elopes, is married, and goes to Santa Fe. Her husband's health fails, and they start, with their two children, overland back to St. Louis. They are overtaken by a snowstorm in the Sangre de Cristo Range. The father dies and the daughter is lost. Ten years later, the wife and son, who had been made members of Major Henning's household, are with him at Fort Advance, in Wyoming. They are attacked by Blackfeet, and the daughter, who had been stolen by a renegade halfbreed, is recovered.

Although this is, in part, a story about a woman and a man falling in love and getting married, it is clearly much more a story of frontier survival. This pointed to the fact that these two genres were not being well understood, which could be addressed in our local best practices, which now provides a list of genre conventions and example novels.

The second way in which the genre classifier has been helpful is in identifying stories that belong to more than one genre. These can often be more difficult to identify, particularly when judging a book by its cover or its series. But genre mashups are of

great interest to researchers, especially those studying the evolution of a genre over time. In the mid-19th century, for example, stories about the frontier and the West were by far the most popular, but this popularity began to give way to more urban tales, often set in New York City and featuring a detective hero. We can read the tension that existed between these genres in our classification errors. Although set in New York and starring one of the most popular dime novel detectives, several issues of the *New York Detective Library* are primarily about the James brothers, who have an association first and foremost with the Western genre. These genre mashups can be more easily identified by a machine, particularly if the classification task is "soft" and probabilities are given for more than one class label.

The next obvious question is whether we might be able to develop a classification model that would be more immediately useful to the cataloger. Instead of simply predicting genre terms, which has limited utility, could we instead predict any Library of Congress Subject Heading (LCSH)? An accurate LCSH model would have the potential to significantly improve the rate of cataloging.

While there are over 257,000 distinct headings in LCSH, catalogers can establish an indefinite number of post-coordinate headings by combining terms according to established patterns. That said, the number of headings can be reduced for the popular fiction of a period, of which only a subset of headings are likely to be relevant. We can safely assume, for example, that historical events that occurred in the 20th century, like World War II, will not come up as a topic in popular fiction from the 19th century. But even then, there remain thousands (or even hundreds of thousands) of headings that might still be relevant. Unfortunately, the number of instances required to train a classifier to accurately assign LCSH terms increases exponentially with the number of classes, which means that a great many novels are required for each subject

heading, especially when multiple subject headings are assigned to the same novels. This makes constructing a classifier for LCSH very challenging.

**Keyword Extraction**

Much like subject headings, keywords are meant to represent the essence of a document or its essential "aboutness," and are a widely implemented part of most information retrieval systems. Keywords are often assigned by indexers, authors, or curators, but just as often there may be no keywords manually assigned to a document. This has given rise to a variety of automatic keyword extraction tools. These tools often focus on statistical analysis of words, comparing word frequency within a text against word frequency within the corpus to determine discriminating words. Other approaches, such as Rapid Automatic Keyword Extraction (RAKE), instead focus on individual documents and extract the same keywords from a document regardless of what other documents might exist in the corpus. This second approach is intended to scale better for a corpus of texts that grows rapidly, because the keywords will not be overdetermined by a corpus in one state. [19] Because *Nickels and Dimes* adds new and diverse titles on a weekly basis, RAKE was the natural choice for experimenting with keyword extraction.

RAKE assumes that keywords usually contain multiple words, but rarely contain punctuation or stop-words, removing stop-words and using phrase and word delimiters to partition a document into a set of candidate keyword phrases. These delimiters include the number of characters within a word, the number of words in a phrase, and the frequency of a word within the document. Co-occurrences of words within these candidate keyword phrases are used to identify the most meaningful phrases, which are scored using the sum of the keyword's member word scores. This is calculated using word frequency, word degree, and the ratio of degree to frequency, where "word

degree" represents how frequently a word co-occurs with other words in the candidate keyword phrases. In other words, the degree is the number of words that occur in the candidate keyword phrases containing the word, including the word itself. After candidate keywords have been scored, the top keywords are extracted, which are defined as the top one third of the words in the document.

RAKE was applied to 252 novels from the same series, *Beadle's Dime Novels,* which had been previously cataloged and digitized. *Beadle's Dime Novels* is often recognized as being the first dime novel series and, unlike many later series, contains novels written by a wide range of authors on a variety of topics, from frontier stories to sea fiction. The program was written in Python using RAKE and the Natural Language Toolkit (NLTK) module, with a character minimum of 5, a phrase maximum of 3 words, and a minimum word frequency of 4 words. Matthew Jocker's stop-word list was not used in this case, because names are one of the aspects of the text that we are most interested in when doing keyword extraction (more on this below); the SMART stop list was used instead, which contains 571 words.[20] In order to account for the sometimes-poor OCR quality, component words were compared against the Unix word list, which contains 69,903 English words and is often used in spellchecking applications. Any keyword phrases that did not at least one word found in the Unix word list were discarded, which was done to account for any OCR noise that might influence results. Because the Unix word list lacks some significant proper nouns, like "Wyoming," care should be used when discarding keyword phrases. Depending on the application, it may be worthwhile to use an expanded list with additional proper nouns before filtering.

[Insert Table 3 near here]

The number of keywords extracted varied significantly from novel to novel, ranging from as few as 4 phrases to as many as 152 phrases. Novels with very few keywords invariably had poor quality OCR, accounting for the relatively low number of results, while the novels with very many keywords were often those that had greater than the usual number of pages (i.e. issues that contained two complete novels or a complete novel and a serialized backup feature). However, these were outliers in the corpus, with only 5 novels in the data set assigned fewer than 10 phrases and only 10 novels assigned more than 100 phrases. Even taking these outliers into account, the average number of keywords per novel was 58 phrases. The number of keyword phrases found can be significantly reduced by increasing the minimum word frequency or by using RAKE's confidence scores, which will discard keyword phrases that fall below a certain confidence threshold.

RAKE excels at uncovering the names of characters, which are usually the first keywords to appear on a list. In general, we only trace a character if they are featured in multiple novels or if they are real world people. For characters like Buffalo Bill, who is well-known and has an established Name Authority record, this is an easy task. But if a cataloger does not trace every character name, it can sometimes be difficult to determine what characters should be traced, because the cataloger will usually have no way of determining in how many novels a character has appeared until all of those novels have been cataloged. This often means that either insignificant characters are traced, in cases where a cataloger traces every character, or lesser known recurring characters are not traced, in cases where only well-known characters are traced. RAKE makes it easier to identify significant characters after digitization by comparing extracted names between novels. That is, we can more quickly identify recurring characters.

Extracted keywords can also be useful when trying to identify groups of people and the context in which they are portrayed. While many dime novels feature Native Americans, for example, relatively fewer feature Native women as anything other than love interests or villains, which makes alternative portrayals especially interesting to researchers. Table 3 shows every keyword extracted from *Malaeska, the Indian Wife of the White Hunter*, which includes "indian woman," as well as words like "house," "child," and "kitchen," which indicates a domestic story. This is a useful discovery, given that there are few domestic stories that are also about Native women.

An obvious shortcoming of automatic keyword extraction is that there is no vocabulary control. Because keywords are not reconciled with a controlled vocabulary, they often contain duplication between synonymous words and phrases. That said, in cases where full-level cataloging with complete subject analysis is not possible, automatic keyword extraction may be an attractive alternative. After all, uncontrolled access points are usually preferable to no access points, especially if the automated nature of the assignment can be effectively communicated to the end user and the number of keywords assigned are limited using confidence thresholds. Efforts can also be made to resolve extracted keywords with existing taxonomies using ontologies, an approach that is examined in more detail below. But the middle ground between fully-automated keyword extraction and full-level subject analysis might simply be to supply catalogers with a list of keywords to aid them in their work.

From a list of keywords, catalogers may be able to infer what those words suggest about the novel. As a tool for identifying abstract concepts or themes, which are often not explicit, keyword extraction may not always be ideal. But one area is which this tool is especially useful is in identifying names, especially of characters, places, and groups of people, which are also often the access points that patrons are most interested

in when they use our popular fiction collections. A cataloger can skim the text of a novel for these words, but if there are more 16 pages, this can be a time-consuming task. Automated keyword extraction simplifies this work, because names have an extremely high likelihood of being identified as significant keywords. It requires much less effort to skim a list of 58 keywords that have been automatically extracted from a text than to hunt for those keywords in a text that might have over 12,000 words.

**Named Entity Recognition**

Named Entity Recognition is a subtask of Information Extraction that involves identifying named entities in a text, such as people, places, and things, and then classifying those entities into categories, such as "person" or "corporate body." When cataloging fiction, Named Entity Extraction is most useful when trying to identify characters and settings. It can be difficult to do this using a dictionary, because names often depend on context for their meaning, e.g. "big apple" might refer to a piece of fruit or to New York City, depending on the context. The main advantage of Named Entity Extraction tools is that they have been designed to take this context into consideration.

DBPedia Spotlight[21] is a tool for automatically identifying names of concepts or entities mentioned in a text and then matching those concepts or entities to unique identifiers. It leverages structured information about more than 4.58 million entities in DBpedia, which contains data extracted from "infobox" tables, categories, and external links in Wikipedia articles. The application uses the Aho-Corasick algorithm to identify candidate phrases through substring matching, with a disambiguation algorithm that is based on cosine similarities and a modification of TF-IDF weights.[22] DBpedia Spotlight is publicly available as a RESTful web service for testing, with the API itself under an Apache 2.0 license, allowing anyone to install and run a dedicated web service.

Unfortunately, confidence and similarity measures only go so far, with wildly inaccurate matches often common when working with these tools. Entities from the 21st century often appear as matches in fiction from the 19th century, in part because contemporary entities are much better represented in Wikipedia. One example is Nick Carter, arguably the most popular American fictional detective in the late 19th and early 20th century, who shares a name with a musician who performed with the Backstreet Boys. Novels featuring the detective will almost inevitably be associated with the musician. For this reason, using a tool like DBPedia Spotlight is especially useful when the cataloger has some idea of the categories of entities they are looking for before analyzing a text. For example, if they are cataloging a series of fiction about the Civil War, like *Blue and Gray Weekly*, they may want to trace military persons and battles that appear in the novels. It is possible to use relevant whitelist filters with DBPedia while annotating a text, such as DBPedia:MilitaryOfficers or DBPedia:Battles, to find more positive matches, which can be narrowed even further by time ranges or particular subjects, such as dbr:American_Civil_War. This significantly reduces the number of entities extracted, but dramatically increases the likelihood of positive matches.

If the cataloger's time is extremely limited, or if assigning keywords automatically, this may be much more desirable than reviewing a longer list of keywords for likely matches. DBPedia can even be combined with keyword extraction, which helps to differentiate between entities that are only mentioned in passing from those that appear prominently in a story. Using RAKE in combination with DBPedia will limit the number of matches found, so that only the most relevant entities are returned, which might be useful when trying to identify historical people in fiction and is also one of the best ways to identify geographic locations in a text.

**Clustering and Topic Modeling**

Clustering and topic modeling are two of the most common unsupervised learning methods, and are popular techniques for the use of classification, document organization, and data visualization. Clustering involves partitioning a collection of documents into groups, or "clusters," based on their similarity to one another. Classes and labels might be supplied to these clusters after the fact, then used to develop a classification model. Topic modeling is a type of "soft" clustering, or probabilistic clustering, that involves assigning "topics" to each cluster. These topics consists of a group of words that frequently occur together, which are meant to reveal some latent semantic meaning hidden in the texts.

The same instances and vectors that were used to develop the classifier described above were used as input in Weka to test Simple k-Means, which is one of the most popular clustering algorithms. Identifying clusters that make intuitive sense is a matter of trial and error, although using a lower k value more often results in clusters that have an obvious relationship to one another. For example, k=2 produced clusters that were equivalent to "Detective and mystery stories" and everything that was not "Detective and mystery stories." Unfortunately, k=7, or seven clusters, did not produce groups that aligned neatly with the 7 genres using to train the classification model described above. "Bildungsromans" was the only distinct genre cluster, with "Detective and mystery stories" split between two separate groups. Every other instance was distributed among the five remaining clusters. See Table 4 for a classes to cluster analysis, which illustrates how the seven genre labels compare to the seven clusters. Analyzing these results could potentially lead to some interesting discoveries; comparing the two distinct "Detective and mystery stories," for example, might aid in narrowing down particular subgenres. But this analysis was out of scope for the current project, which was not meant to be an analysis of dime novel genres.

Topic modeling in MALLET produced similar results. As with clustering, a certain amount of trial and error is involved, adjusting the number of topics and the number of words within the topic until topics are created that make some sort of intuitive sense.

The methods described elsewhere in this paper—assigning genre terms, extracting proper nouns, and supplying a list of keywords—are, in some sense, automatic, requiring relatively little analysis or great familiarity with the dime novel format, literary history, or the 19th century. Except for reviewing large numbers of keywords, most of the subject analysis is being done by the computer, with the cataloger reviewing the results. Clustering and topic modeling, however, are different in that both methods are iterative, requiring trial and error. Classes also need to be assigned to clusters and topics, which requires more effort at analysis, particularly when the relationship between clusters or topics is not self-evident. This is not to suggest that catalogers are incapable of doing this work, but it might be more productive if tools were made available to researchers for analyzing clusters and topics themselves so that they can draw their own conclusions.

## Conclusion

Catalogers do not have to have a firm grasp on the inner statistical workings of each text mining algorithm to work with the tools detailed above. Although some basic familiarity with statistics can be helpful, especially when it comes to tuning models and evaluating results, it is not necessary to understand the math in great depth. The same also applies to computational methods. Although some of the tools discussed require some basic familiarity with at least one scripting language, there are also many tools available that can be readily installed and used with no or minimal background in programming, like Weka and MALLET. Understanding the process of preparing a

corpus, selecting features, and interpreting output is, perhaps, more important.

In the face of ongoing staff reductions, there are plans to begin supplementing the cataloging work that is already being done on the dime novel collections with some of the tools described in this paper. After a novel has been cataloged and digitized, RAKE and DBPedia Spotlight will be used to extract keyword phrases and named entities, which can then be compared against the subjects assigned by catalogers to determine what might have been missed. This will help to ensure that all relevant subject headings have been assigned or to resolve any confusion among catalogers about what subjects are appropriate and how they should be used. While catalogers are still required to perform conventional subject analysis, the hope is that this additional check will allow them to proceed more quickly with less fear of missing something important.

Since the initial experiments described here, *Nickels and Dimes* has grown to over 7,000 dime novels, with plans to add at least another 3,000 volumes over the next two years. As more texts become available, it will be possible to revisit some of the experiments described above. One major shortcoming of our experiments with classification was the limited number of texts in genres like “Love stories” and “Western stories.” Having more instances to work with will not only be a better test of the existing model, but should also aid us in developing more accurate models.

In particular, we are interested in building pairwise linear classifiers to first classify a novel by its genre and then by a subset of relevant subject headings within that genre. Frank and Paynter (2004)[23] worked on a similar project, developing a tool that leverages the hierarchical nature of Library of Congress Classification to automatically assign a classification number to a resource based on its set of Library of Congress Subject Headings (LCSH). They achieved this by training pairwise linear

classifiers to first assign a resource to one of the 21 top-level classifications in the LCC Schedules, then to the most appropriate child node from among more than 4,000 classifications in the LCC Outline. For that project, 800,000 records from the library catalog of the University of California at Riverside were used to train the classifiers, with an accuracy of 55% on a collection of 50,000 resources. Taking a similar approach with NIU's dime novel collection might involve compiling a list of common topics that are often associated with a genre, such as types of crimes for "Detective and mystery stories" or groups of Native Americans for "Western stories," and then training separate classifiers for each genre. After assigning a novel to a genre, the application would then assign to it one or more topical headings. This would likely ignore those topics that appear infrequently, but it may provide a baseline for low-level subject analysis. This could be useful in cases where detailed subject analysis is not feasible due to staffing or other time constraints.

As research advances, as it almost certainly will, there may be further applications for text mining in the cataloging. But for now, text mining can perhaps best be understood as a tool to supplement existing workflows. We are not at a place now, or anytime soon, where text mining will replace the cataloger.

The author wishes to acknowledge Marcos Quezada, Fredrik Stark, and Mitchell Zaretsky, who participated in an experiential learning course in Spring 2016 at NIU that produced the classification application discussed in this article.

---

Notes

[1] *Nickels and Dimes*. https://dimenovels.lib.niu.edu.

[2] Smith, Henry Nash. *Virgin Land: The American West as Symbol and Myth*. Cambridge (Cambridge, MA: Harvard University Press, 1950), 90-109.

[3] Pamela Bedore, *Dime Novels and the Roots of American Detective Fiction* (New York, NY: Palgrave Macmillan, 2013).

[4] Williams, Nathaniel. *Gears and God: Technocratic Fiction, Faith, and Empire in Mark Twain's America* (Tuscaloosa, AL: The University of Alabama Press, 2018).

[5] Pamela Bedore, *Dime Novels and the Roots of American Detective Fiction* (New York, NY: Palgrave Macmillan, 2013), 7.

[6] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," *arXiv:1707.02919 [Cs]*, July 10, 2017, par. 6.2.2. http://arxiv.org/abs/1707.02919.

[7] Ibid. par. 1.1.

[8] Matthew Jockers, "Expanded Stopwords List." http://www.matthewjockers.net/macroanalysisbook/expanded-stopwords-list/.

[9] C. J. van Rijsbergen, S. E. Robertson, and M. F. Porter, "New models in probabilistic information retrieval," British Library R & D Report No. 5587, Computer Laboratory, University of Cambridge, Cambridge, England, 1980. https://tartarus.org/martin/PorterStemmer/.

[10] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," *arXiv:1707.02919 [Cs]*, July 10, 2017, par. 2.1. http://arxiv.org/abs/1707.02919.

[11] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger, "Research-Paper Recommender Systems: A Literature Survey," *International Journal on Digital Libraries,* 17, no. 4 (November 2016): 305–38. https://doi.org/10.1007/s00799-015-0156-0.

[12] Edward T. LeBlanc Memorial Bibliography. https://dimenovels.org.

[13] Matthew Short and Demian Katz, "Linked Open Dime Novels; or, 19th Century Fiction and 21st Century Data" (paper presented at Code4Lib Conference, Philadelphia, 2016). https://2016.code4lib.org/Linked-Open-Dime-Novels-or-19th-Century-Fiction-and-21st-Century-Data.

[14] C.D. Manning, P. Raghavan and H. Schütze. Introduction to Information Retrieval (Cambridge: Cambridge University Press, 2008), 120-125.

https://nlp.stanford.edu/IR-book/html/htmledition/the-vector-space-model-for-scoring-1.html

[15] scikit-learn: Machine Learning in Python. https://scikit-learn.org/

[16] R. K. Anderson, "Smith v. Hancock (1912) and the Death of the Dime Novel" (paper presented at the William M. Blount Symposium on Postal History, Washington, DC, 2006). https://postalmuseum.si.edu/research/pdfs/Anderson.pdf.

[17] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." https://www.cs.waikato.ac.nz/ml/weka/.

[18] Albert Johannsen, *The House of Beadle & Adams and its Dime and Nickel Novels: The Story of a Vanished Literature, Volume I* (Norman, OK: University of Oklahoma Press, 1950), 90.

[19] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley, "Automatic Keyword Extraction from Individual Documents," in *Text Mining: Applications and Theory*, eds. Michael W. Berry and Jacob Kohan (Hoboken, NJ: John Wiley & Sons, 2010), 4. https://doi.org/10.1002/9780470689646.ch1.

[20] S. F. Dierk, "The SMART Retrieval System: Experiments in Automatic Document Processing," in *IEEE Transactions on Professional Communication* PC-15, no. 1 (March 1972), 17–17. https://doi.org/10.1109/TPC.1972.6591971.

[21] "DBpedia Spotlight." https://www.dbpedia-spotlight.org/.

[22]Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes, "Improving Efficiency and Accuracy in Multilingual Entity Extraction," *Proceedings of the 9th International Conference on Semantic Systems (*I-SEMANTICS '13. New York, NY, USA: ACM, 2013), 121–124. https://doi.org/10.1145/2506182.2506198.

[23] Eibe Frank and Gordon W. Paynter, "Predicting Library of Congress Classifications from Library of Congress Subject Headings," *Journal of the Association for Information Science and Technology*, 55, no. 3 (February 2004): 214–227. https://doi.org/10.1002/asi.10360.

| Adventure stories | Bildungsroman | Detective and mystery stories | Historical fiction | Love stories | Sea stories | Western stories |
|---|---|---|---|---|---|---|
| warrior | broker | crimin | colonel | sprung | sailor | prairi |
| trapper | market | disguis | soldier | lover | deck | gulch |
| ranger | stock | hotel | sword | warrior | vessel | calam |
| tribe | illustr | polic | scout | mum | crew | warrior |
| wee | clerk | plot | lieuten | alter | schooner | outlaw |
| fur | sell | crook | warrior | wee | brig | trapper |
| sprung | desk | avenu | confeder | prairi | anchor | rifle |
| savag | rascal | doctor | sprung | nun | pirat | gal |
| rifle | bought | stair | union | god | ashor | miner |
| scout | share | confeder | cano | rifle | cabin | scout |

Table 1. Top 10 signifying words by genre.

| | Adventure stories | Bildungs-romans | Detective and mystery stories | Historical fiction | Love stories | Sea stories | Western stories |
|---|---|---|---|---|---|---|---|
| Adventure stories | 19 | 1 | 1 | 1 | 6 | 0 | 7 |
| Bildungsromans | 0 | 156 | 7 | 1 | 0 | 1 | 1 |
| Detective and mystery stories | 0 | 9 | 121 | 2 | 0 | 2 | 3 |
| Historical fiction | 12 | 2 | 0 | 16 | 2 | 2 | 4 |
| Love stories | 5 | 0 | 0 | 4 | 4 | 1 | 3 |
| Sea stories | 0 | 3 | 2 | 3 | 0 | 16 | 1 |
| Western stories | 6 | 3 | 1 | 2 | 5 | 1 | 36 |

Table 2. Confusion matrix.

| |
|---|
| young girl; madame monet; indian woman; blood; woman; savage; person; strange; world; position; sweet; country; nature; heart; spirit; shadow; house; night; morning; motion; letter; voice; leaves; thing; tribe; strove; embrace; whites; settlement; husband; earth; violence; mouth; death; daughter; child; spring; sunshine; village; sense; garden; comfort; parlor; hemlock; place; cheek; bosom; father; strength; point; speak; water; presence; truth; inter; streets; language; length; remember; margin; track; companion; story; moment; river; kitchen; school |

Table 3. Keywords extracted from *Beadle's Dime Novels* no. 1, *Malaeska, the Indian Wife of the White Hunter*.

| | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 | Cluster 6 | Cluster 7 |
|---|---|---|---|---|---|---|---|
| Adventure stories | 5 | 0 | 88 | 10 | 3 | 0 | 2 |
| Bildungsromans | 12 | 0 | 0 | 0 | 3 | 7 | 452 |
| Detective and mystery stories | 30 | 105 | 4 | 1 | 229 | 39 | 19 |
| Historical fiction | 40 | 0 | 37 | 20 | 0 | 11 | 3 |
| Love stories | 0 | 0 | 29 | 19 | 1 | 1 | 0 |
| Sea stories | 28 | 1 | 15 | 4 | 3 | 0 | 14 |
| Western stories | 83 | 0 | 46 | 17 | 1 | 0 | 5 |

Table 4. Classes to clusters evaluation.