Analysis of Clustering Algorithms to Clean and Normalize Early Modern European Book Titles

Evan J. Bryer[3], Theppatorn Rhujittawiwat[3], Samyu Comandur[3], Vasco Madrid[3], Stephanie M. Riley[2], John R. Rose[3], and Colin F. Wilder[1]

[1]Center for Digital Humanities, University of South Carolina
[2]College of Arts and Sciences, University of South Carolina
[3]College of Engineering and Computing, University of South Carolina

December 29, 2020

1 Abstract

In this paper, we identify the most accurate method of clustering to deduplicate the past centuries book records from multiple libraries for data analysis out of five common algorithms. The presence of duplicate records is a major concern in data analysis. The dataset we studied contains over 5 million records of books published in European languages between 1500 and 1800 in the Machine-Readable Cataloging (MARC) data format from 17,983 libraries in 123 countries. However, each book record was archived by the library owning it. This creates a consistency problem in which the same book was archived in a slightly different way between libraries. Moreover, the change in geography and language over the past centuries also affects data consistency regarding the name of a person and place. Many slightly different names represent the same record. Analyzing such a dataset without proper cleaning will misrepresent the result. Due to the size of the dataset and unknown number of duplicate records with variation, it is impractical to create a lookup table to replace each record. To solve this problem, we use data clustering to deduplicate this dataset. Our work is informed by scholarship on European History and the History of the Book. We find that clustering is an effective method for detecting the slight differences in records caused by the above-mentioned cataloging inconsistencies. Our foundation was experimentation with several candidate clustering methods on a test dataset. The test dataset was prepared by corrupting a clean dataset according to the same characteristics found in the whole dataset. The clean dataset contains roughly 1,000 random records in English, German, French, and Latin with approximately the same language distribution and average record lengths as the whole dataset. Our evaluation reveals that some clustering algorithms can achieve accuracy up to 0.97072. The clustering techniques perform well on the dataset we studied as demonstrated in this paper.

2 Introduction

Scholars in many fields often must works with data that is inconsistent. In Data Science such data is often referred to as “dirty data”. This is especially evident in the case of book title data the further back data sources go, as methods to validate and clean data were far less developed and over time inconsistencies accumulated. Analysis of catalog records across countries and centuries promises to yield significant insight into intellectual and cultural developments around the world, but unfortunately this data often contains duplicates, misspellings, swapped words, and inconsistent spellings.

In this paper we address a particular method for addressing these problems, namely clustering. By creating a test data set to mirror many of the qualities of the bibliographic data we have, we will be able to see how different forms of clustering perform when tested. The test data set allows us to compare the results of the competing clustering algorithms to see which of them is best able to achieve the clustering that matches the original clean data set. To find the accuracy of titles post clustering, we calculated the edit distance percent between the two titles. The accuracy of each method varied quite greatly, with the range of our first test being 15.6 percent, and the best clustering algorithm being able to increase the base accuracy of the test data set by 13 percent.

During the mid-twentieth century, a national, standardized format for cataloging bibliographic information did not exist. Catalogers created and maintained their card catalogs, typing cards for each individual item, a time intensive and expense task, and adhered to local cataloging standards [1]. Efforts to standardized bibliographic information began in the 1950s, but little progress was made until the mid-to-late 1960s.

In 1968, the Library of Congress created MARC II format to standardized bibliographic entries.[1] MARC II became USMARC during the 1980s and then MARC 21 in the late 1990s[3]. MARC records are formatted by three-digit numerically categorized data fields, most of which containing numerous sub-fields with single number or letter designations. The records contain bibliographic data about the books and none of the actual content of the books, serving as a means of storing metadata about the books. Our study specifically works with the MARC 21 standard and the data set consists of data serialized in MARCXML entries. MARCXLM is an XML schema specifically designed around MARC 21, the modern rendition of the format that came about due to the combination of the US and Canadian MARC formats.

[1] In 1965, sixteen libraries participated in the Library of Congress’s MARC I pilot project. The success of MARC I led to the creation of MARC II.[2][3]

In 1967, Online Computer Library Center (OCLC) signed their Articles of Incorporation as nonprofit computer library network, with the mission to help colleges and universities further scholarship and education by providing affordable library resources[4][5]. OCLC launched its shared cataloging system, OCLC Union Catalog (now known as WorldCat), on August 26, 1971.[2] OCLC streamlined library cataloging by implementing national standards for how catalogers entered bibliographic information and by eliminating the need for more than one library to catalog an item. As of July 2020, approximately 18,000 libraries in 123 countries participate and use OCLC services, including WorldCat, which contains over 490 million records in 483 languages from over 3 billion library holdings[7][8]. OCLC uses both human and machine resources to clean, monitor, enhance, and correct bibliographic records in WorldCat.[3] While bibliographic standards exist, the sheer number of records coming from countries around the world makes consistency a challenge.

Our main data set was procured through an agreement between our research team, led by Colin Wilder, and OCLC on October 31, 2019. We were given use of it for scientific study for a three-year period. The principal two restrictions on our activity are that we cannot make commercial use of the data nor publicize the records themselves other than as occasional individual samples. Neither of these restrictions affect our study.

Next, there are two main problems in processing and studying the data set. First, its sheer scale, at about 6 million records. Second, there are a very large number of dirty or duplicative records. Over 300,000 records were found by our methods to be duplicate entries, and over 3 million titles appeared in more than one form throughout the data set.

Since, as discussed above, the set is 6 million records in size and is an undeduplicated union set from over 10,000 libraries around the world, it was to be expected that the cataloging language would show great variance, with spellings of country names, language names, city names, and even author names *for the same resource* varying among duplicate records, to say nothing of misspellings. While the scale is simple enough to deal with, just a matter of writing more time efficient code when possible, and giving the processes adequate time to run, the dirty nature of the records are a much more difficult problem to address. Cleaning the data, normalizing the spellings and formats of entries, as well as accounting for differences between languages are invariably going to be crucial to getting any kind of valid results from analyzing the data set in historical context, otherwise large amounts of valuable data could be missing when queried.

To generalize different spellings for names, locations, and indeed all other words, the straightforward approach is creating a lookup table which contains

[2] OCLC began accepting member-input entries on October 18, 1971. In 1997, OCLC Online Union Catalog changed its name to WorldCat.[6]

[3] The Enhance Program, the CIP (Cataloging-in-Publication) Upgrade Unit, the Program for Cooperative Cataloging (PCC), the WorldCat Quality Team, and the Duplicate Detection and Resolution (DDR) software are a few of programs and initiatives OCLC uses to enhance records. The WorldCat Quality Team enhanced 2,407,575 records in July 2020 and 69,668,253 records between July 2019 and July 2020.[4][5][9]

the mapping of all possible spellings to general spellings. This approach is simple but proved to be difficult for this data set due to the size and diversity of this data set. Manually creating a lookup table for this size of data is difficult and time consuming. Many researchers have proposed clustering algorithms to classify books [10]. Most research focused on the task of clustering books from a recent time period, generally with only one or two languages represented. However, our data set contains varieties of books from 1500-1800 in Europe in many different languages. Additionally, the orthography (spelling) of these languages underwent significant change over that time period, so in a sense there are far more languages than only the nominal ones (English, Dutch, French, German, Italian, Latin, and Spanish), in a sense there are many more or even a miasma of different languages because of the widely varying spelling and orthography, both synchronically and diachronically, if one accounts for this longitudinal variance We experimented clustering on our data set and received a satisfied result.

In this paper, we describe a clustering algorithm that exhibits superior performance on the OCLC data set. While the methods we test and conclusions we draw we are primarily focused on the computer science aspects of this problem, the ramifications of our analysis will also affect the field of bibliography as the field often handles MARC records. With this knowledge, the efforts both fields make when cleaning and normalizing MARC records can be mitigated.

3 Related Work

Working with “dirty” MARC records is far from a new problem both in the fields of computer science and bibliography. Many researchers in the past have taken different approaches to accomplish this from using forms of data cleaning and data harmonization, normalizing alternate spellings and punctuation, on their data sets to creating completely new clustering algorithms to compete against the existing methods. While this paper is primarily focused on the computer science aspects of handling MARC records, the information we have gathered and tests we have run should prove equally pertinent to bibliographers, giving insight into methods to approach handling diverse types of bibliographic data. For computer scientists, the use of clustering on a multilingual data set can provide insights about the restrictions of certain types of data clustering, as well as the situations in which different types are able to perform the best. Of course, we are far from the first to tackle this problem, and as such we would like to acknowledge the past achievements and developments made by scholars as well as the impact they had on the field as well as this paper.

Mikko Tolonen and his colleagues have explored data cleaning quite extensively through their study of bibliography. In Tolonen, Mikko, et al [11], they used cleaning in a way much like we do, taking the information such as titles and authors and attempting to fix any errors within them. While this is an inspiring achievement, we were unfortunately not able to apply their approach because their article does not significantly document how they performed this, as it is not the focus of their paper. Tolonen, Mikko, et al. [12] also used some novel methods of cleaning their data. Their paper describes their usage of string distance ranking algorithms and using additional information such as number of pages to rank the matches. Our usage of clustering as a means of cleaning not only addresses the forms of cleaning that these papers took, but also addresses harmonization. This is a problem tackled in many papers. A great example is the work by Marjanen, Jani, et al. [13] in which they normalize special characters and spelling errors. They even go a step beyond simple clustering and normalize synonymous words to lower the variability of titles.

Those in the field of Computer and Data Science often find the need to handle dirty data. Despite the numerous methods of cleaning and normalizing data, few can do both with the accuracy of clustering in data sets such as ours, and as such there are many papers exploring the forms and usage of this approach. Some of these papers use the vast amount of readily available bibliographic data to test their developments. An example is the work of Li, Jingxuan, and Tao Li [14]. They not only address existing forms of clustering algorithms but create their own algorithms and test them against existing methods. In their paper, they use an example data set consisting of books and keywords. Their method, called Hierarchical Co-Clustering (HCC), brings together two related but different themes from clustering: hierarchical and co-clustering. Both themes have differing goals; hierarchical clustering enables browsing and navigation, while co-clustering clusters different types of data at the same time by using information about their relationship. HCC begins with singleton clusters and merges the two closest clusters until only one remains, all while using the agglomerative hierarchical clustering algorithm as the framework. By comparing the HCC algorithm to more common techniques such as K-Means and Single Linkage Hierarchical Clustering (SHLC), they can show how much more efficient their method is. Similarly trying to improve on the methods of clustering, Waluyo, Arif Mardi, Eko Prasetyo, and Arif Arizal. [10] create a method of clustering book titles based on the K-Means average of distance dissimilarity. The method this paper uses produced great results for categorizing books based on shared features. However, the problem they sought to solve required far fewer clusters than title normalization. The method can effectively group books into categories such as law, economics, accounting, engineering, management, and information technology, as shown in their tests utilizing a data set of 500 book titles from the Library of Bhayangkara Surabaya. Péter Király [15] attempted to validate the metadata of records, ensuring that linked titles existed in records, that ISBN and ISSN fields had proper identifiers, and that the control field was properly formatted. There were many great insights about the problem of “dirty” MARC data within this paper, focusing primarily on classifying records as valid or invalid.

The field of Machine Learning has also had an impact on this area. Ozsarfati, Eran, et al. [16] assess different machine learning algorithms to solve the problem of genre classification for books based on their titles, similarly to our paper’s goals of comparing different algorithms and their accuracy of grouping similar titles. The Machine Learning techniques used in this experiment are

Naive Bayes, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Bi-Directional LSTM (Bi-LSTM). This paper provides experimental results on 207,575 samples of book titles from 32 different genres. Much like Péter Király's paper [15] however, the number of categories is far lower than what would be needed for the use of normalization. This is a consequence of their focusing on a much different problem. Bleiweis [17] presents a model capable of combining a word vector set of a varying number of elements in order to represent a verse or a sentence, with an iterative distance metric that evaluates similarity in pairs of non-conforming verse matrices. The work in this paper investigates linguistic structure in raw texts and uses Word2Vec to explore clustering and search tasks to train the underlying word representations. Rather than reshaping the dimensionality of input sentences into uniform feature vectors as other techniques do, the model in this paper retains the word vectors as separate rows in a matrix form in order to construct any of the verse, chapter, or book data structures to complete linguistic tasks. The proposed formulation improves the performance of semantic clustering and searching, which is extremely valuable for a task as time intensive as clustering.

4 Data Set Cleaning

We were provided with the data set in a single large XML file containing approximately 5.8 million records of data. These records followed the traditional MARC21 tag structure, with control fields providing a summary of the bibliographic information, followed by details about the title, author, edition, physical description, and other related works. A MARC field (a three-digit numeric field) has a set of subfield codes associated with that field (a one-character alphanumeric field).

The XML tree structure of the data was initially parsed into an equivalent JSON file for ease of access, and then this JSON file was split into a set of TSV files. The TSV files were split by MARC field. Each column of a TSV file is a MARC subfield. The use of a table structure allowed for easy conversion to a relational database, and the data was imported into SQL for access and management. In addition to clustering, basic cleaning methods for removing punctuation marks, accent marks and other special characters, and capitalization were employed to standardize the data.

5 Clustering

While no algorithm could perfectly address the problem of producing a correct book title from a potentially inaccurate title, clustering provides a method of bringing the entire set of data to a more accurate, more harmonized state overall. Due to the fact our data set contains many duplicates and near duplicates, as well as very few titles and authors that only appear once in the OCLC data set, using clustering as a cleaning method fit the task well. Even more so than cleaning, clustering will allow for the harmonization of the data, which is also valuable as it aids in the querying process of the data further down the line. Because the time period that this data set covers is so vast, and includes many different countries, there will be many cases where a word or name will be spelt differently depending on where the publication record originated. As such, methods that focus on addressing misspellings and typos in the publication records may leave multiple different forms of names or words throughout the data. The harmonization that clustering allows for addresses this problem.

To keep any quality differences between different libraries for clustering from affecting the analysis, we will be using a single program that allows for the use of multiple different clustering techniques throughout the analysis: Openrefine. We will be addressing five main methods of clustering, falling under the categories of Key Collision and Nearest Neighbor. These five categories will be Fingerprint, N-Gram Fingerprint, Phonetic Fingerprint, Levenshtein Distance, also called edit distance, and Prediction by Partial Matching, also known as PPM. The idea behind Key Collision is to create keys to represent the most significant parts of the string, where Nearest Neighbor takes the more traditional approach to clustering, allowing for the user to choose a distance threshold for strings to be grouped.

Each Key Collision method works slightly differently in their process of creating the keys. Fingerprinting is the most straight forward, creating the key by removing additional white space, changing all characters to lowercase, removing punctuation and control characters, normalizing extended western characters to their ASCII representation, splitting the string into whitespace-separated tokens, sorting the tokens and removing duplicates, and then joining the tokens back together. N-Gram fingerprinting works very similarly, except before the string is separated into tokens, the n-grams are gathered. Doing this allows for small differences between words that would not have been picked up by the original fingerprinting method to be accounted for. Finally, the phonetic fingerprinting generates a key by transforming the tokens into their phonetic representations.

The main difference between the two forms of Nearest Neighbor clustering is the method in which each string is assigned a value. Levenshtein Distance is a very traditional method of assessing the difference between two strings, calculating the edit distance, or number of edits required to change one string into the other. This form of clustering is generally especially good for detecting typos, as a single incorrect letter, or pair of swapped letters, will give a very small edit distance. Prediction by Partial Matching, on the other hand, uses the idea of Kolmogorov complexity to estimate the similarity between strings [18]. Text compression generally works by creating shortened ways to represent repeating sections of text [19]. Because of this, if you have two strings, string A and string B, if A and B are identical then the compressed string AB compressed will be almost identical to the compressed versions of A or B on their own. Using this logic, Openrefine calculates the distance between strings with the formula $d(A,B) = comp(A+B) + comp(B+A)/(comp(A+A) + comp(B+B))$, where comp(s) is the length of bytes of the compressed sequence of the string s and + is the append operator.

To test each clustering option's accuracy in removing misspelled titles and other corruptions that the titles present with, we needed to be able to use a data set that reflects many of the important attributes of the OCLC data set, but had a base truth that did not exist in OCLC's. Creating a set of manually vetted data from the OCLC data set initially seemed like the most apparent method of solving this, but the issue of scale made this approach impractical. The main reason for this was in the selection of a subset, where too small wouldn't reflect the language distribution nor the repetition of titles accurately due to the sheer size of the entire data set, but too large of a selection would take an infeasible amount of time only further extended by the unfamiliarity of many of the languages to our team. As such, a mock data set was constructed to mirror the qualities of the original data as closely as possible, with the benefit of an existing base truth. While it is very possible that the use of a mock data set will leave out some forms of errors that exist in the original set, it provides a feasible method to see how each clustering method performs on a large scale for data of the same format.

To create this data set, we took roughly 1000 proverbs and idioms from English, German, French, and Latin to mirror the approximate average length of book titles found in the OCLC data set, as well as including a distribution of languages found in the data set as to not skew the results in the case of a clustering algorithm working particularly well on one language but not others. This concern is primarily focused on the phonetic clustering algorithms, for which using only one language could drastically alter the accuracy in ways that would not be reflected in the OCLC data set. Using proverbs and idioms allowed for great variation in the length of individual entries, while still having an average length similar to that of many titles. The use of such phrases also did not pose a problem for our intended use of clustering on the original data, as we had not intended to cluster based on anything other than just the title data. This is both because of the restrictions of Openrefine, which is only capable of clustering by one attribute of inputted data at a time, as well as author data not being required for a record to be valid. The absence of such data in many records could cause titles that should be clustered together to be skipped over by the algorithm.

To make clustering a viable method of cleaning on this test data set, we created two duplicates of the entries, each one corrupted in some fashion. The corruptions we selected to apply were based on both our observations of our data set, as well as those of other scholars on similar sets of OCLC data [20]. These corruption are incorrect letters, letters swapped with their neighbors and words swapped with their neighbors. For each letter in the first duplicate set, there was a 7 percent chance of it randomly being swapped with any random letter. For the second duplicate set, there was a 10 percent chance for the random letter swapping, as well as a 6 percent chance for the current word to be swapped with adjacent words. These percentages were chosen due to most errors in the data set being relatively minor in scale. As such, we wanted only one or two “corruptions” per title, so such low percentages for corruption rate proved to mirror this well in the artificial data set. These two sets, as well as the original set, were then concatenated into a single new line delimited text file. This served to imitate the propensity for entries to be duplicated, and for duplicate entries to have minor differences from the original in the OCLC data set. To test the post-clustering data, we created a set to be considered the ”correct” set, made up of the uncorrected titles in the same order as the test data set. The first test we performed was to evaluate the percent of titles that were correct. The corrupted data set had a 42.1 percent similarity to the correct set, with the following being the percent accuracy after each form of clustering was performed.

Full String Matching	Accuracy
Fingerprint	0.421
Ngram-Fingerprint	0.419
Metaphone3	0.420
Cologne-Phonetic	0.425
Daitch-Mokotoff	0.395
Beider-Morse	0.423
Levenshtein (r = 1)	0.436
PPM (r = 1)	0.473
LevenshteinAlt (r = 2)	0.479
PPMAlt (r = 2)	0.551

Table 1: Clustering results based on full string matching

The next test we performed was similar in concept, but instead of matching the entire strings, we performed a string similarity calculation on it. The similarity calculation was performed by taking the Levenshtein distance between the original title and the title after being put through the clustering algorithm. The original corrupted data set had a predictably high similarity of 95.8 percent and the following being the post clustering results.

String Similarity	Accuracy
Fingerprint	0.958
Ngram-Fingerprint	0.958
Metaphone3	0.956
Cologne-Phonetic	0.958
Daitch-Mokotoff	0.933
Beider-Morse	0.957
Levenshtein (r = 1)	0.958
PPM (r = 1)	0.961
LevenshteinAlt (r = 2)	0.961
PPMAlt (r = 2)	0.971

Table 2: Clustering results based on string similarity

Given these results, the best form of clustering out of those assessed for the purpose of this data set appears to be Prediction by Partial Matching using a radius of 2. In many situations, this form of clustering has the drawback of overreaching and adding many false positives to clusters. Due to the relatively long average length of book titles, this problem is a lot less apparent. This, however, may not be the case in the OCLC data set. The test data set was made to approximate the length of book titles, but many real world titles are between one and five words, in which case the overreaching of Prediction by Partial Matching could become much more serious of an issue. Predictably, some of the worst performing algorithms were those based on phonetics, as such an approach would be much better suited for data sets consisting of only a single language.

The problem of overreach by these algorithms are able to be assessed by the number of false positives each algorithm generates. For the purposes of this analysis, a false positive will be defined as a string that is correct in the corrupted data set, but incorrect after clustering and false negatives are strings that should have been corrected and were not. Because of how the data set was constructed, that means nearly all of these false positives will land in the first third of the data set, where no corruption was performed on the strings. After calculating the number of false positives, we found the following.

False Positives and Negatives	Positives	Negatives
Fingerprint	8	1745
Ngram-Fingerprint	7	1752
Metaphone3	272	1485
Cologne-Phonetic	73	1667
Daitch-Mokotoff	404	1428
Beider-Morse	14	1734
Levenshtein (r = 1)	204	1502
PPM (r = 1)	320	1275
LevenshteinAlt (r = 2)	328	1248
PPMAlt (r = 2)	440	920

Table 3: False Positives and Negatives by Clustering Type

As was expected, the clustering methods that generated the broadest clusters also had the greatest number of false positives. Surprising, however, was PPM and Levenshtein's ability to have an overall greater accuracy despite having some of the highest false positive counts. The other form of clustering that gave a large number of false positives, Daitch-Mokotoff, had an overall lower accuracy than the original data set, which would be expected of such high false positives. This result is easily explained when examining the number of dirty records that were caught and fully corrected by the algorithms, however.

Corrected Strings	Count
Fingerprint	8
Ngram-Fingerprint	1
Metaphone3	268
Cologne-Phonetic	86
Daitch-Mokotoff	325
Beider-Morse	19
Levenshtein (r = 1)	251
PPM (r = 1)	478
LevenshteinAlt (r = 2)	505
PPMAlt (r = 2)	833

Table 4: Data Points Cleaned

Given these counts, we can easily calculate the precision and recall of each algorithm.

Precision and Recall	Precision	Recall	F-Score
Fingerprint	0.5	0.005	0.010
Ngram-Fingerprint	0.125	0.004	0.008
Metaphone3	0.496	0.153	0.234
Cologne-Phonetic	0.541	0.049	0.090
Daitch-Mokotoff	0.446	0.185	0.262
Beider-Morse	0.576	0.011	0.022
Levenshtein (r = 1)	0.552	0.143	0.227
PPM (r = 1)	0.599	0.273	0.375
LevenshteinAlt (r = 2)	0.606	0.288	0.390
PPMAlt (r = 2)	0.654	0.475	0.550

Table 5: Precision and Recall

Given our results, using Prediction by Partial Matching clustering with a radius of two generated the best overall results, with the highest recall and precision, as well as the largest increase to accuracy.

6 Conclusion

The OCLC data set, despite being extremely valuable in the information that it contained, was plagued by duplicate entries, misspelled titles, swapped words, and words whose spellings differed over the years. To make the best use of this data as possible, and for the information we extrapolate from it to be as accurate as possible, it was a necessity to clean up the data to make queries far more representative of the truth of the data set.

Due to the vast majority of the books appearing multiple times throughout the data, one of the most apparent solutions to tackle many of the problems presented at once was the use of clustering data. Because the data we have represents multiple languages throughout hundreds of years and includes multiple forms of words and titles as they appeared during this time period, the use of simple spell checking algorithms would have been far less powerful on this set. Clustering has been around for many years and as such exists in many forms, with the accuracy and use cases of each varying drastically based on what they were designed for. To assess how they would compare against one another on the OCLC data set, we constructed a test data set with many of the same attributes as the OCLC data and analyzed the accuracy of each clustering method both in percent of titles fully corrected as well as the overall accuracy of the set compared to the base set.

By comparing a few of the most often used clustering methods, we were able to determine the best clustering algorithm for the use case of cleaning book titles, particularly in a data set with many repeating titles and multiple languages throughout. While the statistics and figures gathered were generated from the use of a test data set, the characteristics of the test data set was designed to mirror those of the OCLC data set to make the results easily extrapolated to it. Many of the results we assessed fit the predictions for them, such as phonetic clustering being particularly poorly suited for a data set as diverse in language as the OCLC data set and Nearest Neighbor clustering producing generally superior results at the cost of processing time and complexity. Showing the biggest improvement both in the percent of fully corrected strings, as well as the greatest increase in the overall accuracy of the set, PPM with a radius of 2 was the best suited clustering algorithm for our case. The radius, however, may not be the perfect fit for all areas of the data set, as such a relatively large radius could present over fitting errors for short titles.

Future papers could expand on our research both by testing different clustering algorithms against the results we gathered from the algorithms we tested as well as by developing new data cleaning algorithms designed specifically for book titles. The first method is straightforward, there are dozens of clustering algorithms that could be applied to this data set and many have the potential to outperform the best algorithm that we found. A much larger undertaking, designing a novel cleaning method for book titles could be a highly valuable expansion of this research. One potential solution would be performing the PPM clustering but checking the cluster values against a database of book titles, searching for the closest match and if its above a specific threshold, applying it instead of the value PPM. This could vastly improve the accuracy of the algorithm by cutting down the number of clusters that have the incorrect value normalized onto them.

References

[1] L. Leighton, "Changing the tasks of cataloging," *Journal of library administration*, vol. 25, no. 2-3, pp. 45–54, 1998.

[2] *Library of congress, frequently asked questions (faq)*, Nov. 2020. [Online]. Available: `https://www.loc.gov/marc/faq.html#marc21vsuscan`.

[3] M. Seikel and T. Steele, "How marc has changed: The history of the format and its forthcoming relationship to rda," *Technical Services Quarterly*, vol. 28, no. 3, pp. 322–334, 2011.

[4] K. W. Smith, "Oclc: Yesterday, today and tomorrow," *Journal of library administration*, vol. 25, no. 4, pp. 251–270, 1998.

[5] J. Jordan, "Oclc 1998–2008: Weaving libraries into the web," *Journal of Library Administration*, vol. 49, no. 7, pp. 727–762, 2009.

[6] P. Schieber, "Chronology: Noteworthy achievements of the cooperative 1967–2008," *Journal of Library Administration*, vol. 49, no. 7, pp. 763–775, 2009.

[7] *Oclc technology*, Nov. 2020. [Online]. Available: `https://www.oclc.org/en/technology.html`.

[8] *Inside worldcat*, Nov. 2020. [Online]. Available: `https://www.oclc.org/en/worldcat/inside-worldcat.html`.

[9] *Oclc delivers quality*, Nov. 2020. [Online]. Available: `https://www.oclc.org/en/worldcat/cooperative-quality.html`.

[10] A. M. WALUYO, E. PRASETYO, and A. ARIZAL, "Clasification system of library book based on similarity of the book title using k-means method (case study library of bhayangkara surabaya)," *JOURNAL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, VOL 3 NUMBER 1, JUNE 2018*, vol. 3, no. 1, 2018.

[11] M. Tolonen, L. Lahti, H. Roivainen, and J. Marjanen, "A quantitative approach to book-printing in sweden and finland, 1640–1828," *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, vol. 52, no. 1, pp. 57–78, 2019.

[12] M. Tolonen, J. Marjanen, H. Roivainen, and L. Lahti, "Scaling up bibliographic data science.," in *DHN*, 2019, pp. 450–456.

[13] J. Marjanen, V. Vaara, A. Kanner, H. Roivainen, E. Mäkelä, L. Lahti, and M. Tolonen, "A national public sphere? analyzing the language, location, and form of newspapers in finland, 1771–1917," *Journal of European Periodical Studies*, vol. 4, no. 1, pp. 54–77, 2019.

[14] J. Li and T. Li, "Hcc: A hierarchical co-clustering algorithm," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 861–862.

[15] P. Király, "Validating 126 million marc records," in *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, 2019, pp. 161–168.

[16] E. Ozsarfati, E. Sahin, C. J. Saul, and A. Yilmaz, "Book genre classification based on titles with comparative machine learning algorithms," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, 2019, pp. 14–20.

[17] A. Bleiweiss, "A hierarchical book representation of word embeddings for effective semantic clustering and search.," in *ICAART (2)*, 2017, pp. 154–163.

[18] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi, "The similarity metric," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '03, Baltimore, Maryland: Society for Industrial and Applied Mathematics, 2003, pp. 863–872, ISBN: 0898715385.

[19] S. Shanmugasundaram and L. Robert, "A comparative study of text compression algorithms," *ICTACT Journal on Communication Technology*, vol. 2, Dec. 2011. DOI: `10.21917/ijct.2011.0062`.

[20] E. T. O'Neill, S. A. Rogers, and W. M. Oskins, "Characteristics of duplicate records in oclc's online union catalog," 1993.