Using Text Mining Tools to Inform Search Term Generation: An Introduction for Librarians

Bethany S. McGowan

abstract: The use of text mining tools can help librarians improve the precision of searches, increase search sensitivity, and translate search strategies across multiple research databases. When combined with the intuitive approaches that librarians commonly use, text mining tools help reduce biases by improving the objectivity, transparency, and reproducibility of search strategies. This paper seeks to boost librarian confidence by providing a step-by-step guide for conducting text analyses with two free text mining tools, Voyant Tools and the statistical programming language R. It provides a brief introduction to concepts related to the use of text mining tools for search term generation, then details processes for using Voyant Tools and R and RStudio as investigative instruments.

Background

Systematic reviews and similar evidence syntheses—such as iterative reviews and metanalyses—provide a summary of the available evidence on a topic by synthesizing the results of numerous individual studies. This synthesis provides one of the highest levels of evidence and is relied on to inform decision-making and to assess the effectiveness of interventions. Due to an expectation that systematic reviews be comprehensive and reproducible, recommendations have been established to guide the conduct and reporting of them.[1] The *Cochrane Handbook for Systematic Reviews of Interventions* is a popular example. Highlighted in chapter four of the *Cochrane Handbook* is the requirement for review teams to work with a librarian: "Review authors should work closely, from the start of the protocol, with an experienced medical/healthcare librarian or information specialist."[2] Review teams that include librarian coauthors cor-

portal: Libraries and the Academy, Vol. 21, No. 3 (2021), pp. 603–618.

relate with higher quality reported search strategies.[3] In addition, librarians who join systematic review teams build stronger ties with researchers, gain a deeper understanding of faculty research practices, and may coauthor major publications.

Systematic reviews require many hours of labor, however, for a final product that may take months or years to reach publication; or, unfortunately, a product that may never be made available. One study found that it took an average of 67.3 weeks for systematic reviews registered in the Prospero registry (International Prospective Register of Systematic Reviews) to be published; funded reviews took even longer.[4] While systematic reviews have been popular in health sciences for many years, and health sciences librarians have a long history of supporting systematic review services, such reviews have also become popular in other disciplines.[5] As the demand for systematic review services has grown among researchers across academic disciplines, subject librarians must understand the potential for text mining to help optimize their investment in such services.

As the demand for systematic review services has grown among researchers across academic disciplines, subject librarians must understand the potential for text mining to help optimize their investment in such services.

Text mining is an automatic process to extract insights from large amounts of text, aided by software that can identify concepts, patterns, topics, keywords, and other characteristics. Text mining tools support the development of search strategies by revealing high-frequency terms, uncovering correlated or collocated words—that is, words that often occur together or in close proximity—and highlighting differences in vocabulary use across research databases.[6] Such tools increase the transparency, reproducibility, and objectivity of the search strategy development process by providing a means of quantifying, validating, and documenting term selection.[7]

In addition to optimizing search strategy development, text mining tools help address some common criticisms of the intuitive search development process—that it is biased, subjective, not transparent, and not reproducible.[8] Such faultfinding is particularly troublesome because systematic reviews rely on users easily accessing and reproducing their results as fundamental methodological tenets. Claire Stansfield, Alison O'Mara-Eves, and James Thomas list intuitive search development aids as "knowledge of the literature, published preexisting searches in related areas, topic expertise, database thesauri, iterative searching, [and] browsing citations within databases."[9] Other practices include checking reference lists, citation mapping, contacting potential informants, reviewing conference proceedings, and consulting gray literature, such as preprints and other unpublished material.[10] To reduce biases in search strategy development, librarians rely on peer critique and discussions with members of the review team. Still, critics argue that search strategy development remains prone to biases. Search term generation informed by text analysis is often highlighted as an objective means of search strategy development. However, using solely text mining tools for search term generation does not identify parallel terms and reinforces biases by resulting in the return of similar studies.[11] Therefore, a blended approach of intuitive search development paired with the use of text mining tools could provide a more objective search strategy and retrieve more relevant results.

Text mining tools help members of the review team analyze patterns in text, discover relevant words and phrases, and minimize the retrieval of irrelevant studies. Methodological approaches for using text mining tools to identify studies are well documented.[12] However, the actual process of using such tools is less well reported. This paper seeks to provide a step-by-step guide for librarians seeking to learn to use text mining tools. It provides a brief introduction to concepts related to the use of text mining tools for search term generation, then details the processes for using Voyant Tools and R or RStudio. This work addresses the following research question: How can text mining tools be used to reduce the biases associated with intuitive search development approaches?

. . . intuitive search development paired with the use of text mining tools could provide a more objective search strategy and retrieve more relevant results.

Information Retrieval with Text Mining Tools

Regular and substantial increases in the amount of published literature and improvements in the efficiency and accessibility of search engines make it possible to find large quantities of information. Conducting a systematic review can feel overwhelming, as researchers are expected to find all the literature related to a research question. Thankfully, advances in information technologies, especially progress in text mining tools, can make the identification of relevant studies easier. Text mining tools help extract key terms from documents and allow for the identification of associations among words and phrases, which could lead to new and unexpected discoveries.

Text mining tools help extract key terms from documents and allow for the identification of associations among words and phrases, which could lead to new and unexpected discoveries.

Text mining tools support three main outputs:

1. Information retrieval, or the gathering of text related to a specific query;
2. Information extraction, or the identification and distillation of snippets of text related to a query; and
3. Data mining, or the recognition of direct and indirect associations among information extracted from text data.[13]

These outputs of text mining can be correspondingly mapped to systematic review processes:

1. Retrieval of studies related to a specific query, which maps to information retrieval;
2. Describing characteristics of included studies, which maps to information extraction;
3. Identifying patterns in included studies, which maps to data mining.[14]

Choosing the appropriate keywords is crucial for search development and directly influences the identification and retrieval of relevant studies.

Angela Spencer and Jonathan Eldredge list search development as one of the most documented roles for librarians on systematic review teams.[15] Choosing the appropriate keywords is crucial for search development and directly influences the identification and retrieval of relevant studies. However, it can be difficult to single out search terms, especially when faced with such challenges as diverse terminology, varied vocabularies, a range of conceptual perspectives, and inconsistent indexing. Text mining tools optimize search term development by:

1. Improving search precision and sensitivity by identifying frequently used terms, phrases, and word combinations; and
2. Translating search strategies across multiple research databases by identifying free-text terms—that is, alternate language that might appear in relevant sources—from studies that would not have been included if using only controlled vocabulary, and vice versa—identifying controlled vocabulary terms from studies that would not have been included if using only free-text terms.[16]

Creating an Objectively Derived Search Strategy

Elke Hausner, Siw Waffenschmidt, Thomas Kaiser, and Michael Simon's "Routine Development of Objectively Derived Search Strategies" introduces a four-step process that increases transparency and efficiency in the selection of candidate terms for search strategy development.[17] It establishes an "empirically guided approach to the development of a search strategy."[18] The four-step process includes the generation of a test set, the development of a search strategy, validation of the strategy, and standardization of documentation for the search development process. The following is a modified approach to the Hausner team's process.[19]

Generating a Test Set

Create a test set of relevant references by conducting a preliminary search for each database. Use intuitive search development strategies to design the preliminary search. Select several obviously relevant references, working with other members of the review team if needed. These references will form a test set, which represents a "quasi-gold standard" of pertinent sources for the research question. In their example, Hausner and her team created a test set of 38 relevant references.

Randomly divide these references. Two-thirds will serve as a development set, and the remaining one-third will become a validation set.

Developing Objective Search Strategies

Identify free-text terms and controlled vocabulary terms. Free-text terms are alternative words for key concepts, including synonyms, other spellings, abbreviations, and changes in terminology over time. Detailed instructions for using both Voyant Tools and the statistical programming language R to generate free-text terms and controlled vocabulary are available in the "Voyant Tools" and "R and R Studio" sections of this article.

Manually assemble the identified free-text and controlled vocabulary terms into a search strategy. When compiled, the results of the final search should capture every reference from the development set while remaining precise enough to exclude an overwhelming number of irrelevant results. Consult with the review team to adjust the search as needed.

Validating the Search Strategy

Use the validation set—that is, the one-third of sources not included in the development set—to confirm that the final search strategy works with an alternative set of references. Every reference from the validation set should be captured in the final search strategy.

Standardize the internal documentation. To further support transparency and reproducibility, carefully document the decision-making and results of the previous three steps.

Voyant Tools

Voyant Tools is a free, open-source, Web-based text analysis application that provides an interactive means to execute basic text mining.[20] Benefits of Voyant Tools include a low barrier of entry that allows new users to quickly execute and visualize text analysis. It also runs on most browsers with no downloads or logins required and can work with data in multiple formats, including plain text, pdf (portable document format), XML (extensible markup language), and RTF (rich text format).

Voyant Tools is a free, open-source, Web-based text analysis application that provides an interactive means to execute basic text mining.

Voyant Tools is ideal for exploring terms and their connections. It automatically creates graphs and generates visualizations such as word clouds, images composed of words in which the sizes indicate their frequency.[21] A disadvantage is that Voyant Tools has preprogrammed functionality which can be limiting. In addition, its interactive interface creates visualizations that are ideal for individual exploration but less suited as outputs for scholarly sharing.

Creating a Data Set

Follow the steps in the "Generating a Test Set for Each Database" section to create a development set. Import the references from the development set into a citation management tool. The text from these references will serve as the data for the text analysis. Export the references as a file from the citation management tool, including the title, abstract, and keyword data. Voyant Tools will recognize a range of file types, including plain text, HTML (HyperText Markup Language), XML, MS Word, RTF, and pdf.

Consider analyzing the full text. An abstract analysis will provide a broad view of major themes that will help identify free-text terms and controlled vocabulary, but a full-text analysis will allow for deeper exploration that can reveal correlated terms and associated words. Create a file for full-text analysis by copying the entire text of each article from the development set into a single, comprehensive text file. Ensure that keywords and controlled vocabulary terms are included. Create a separate text file for each database searched.

Creating a Corpus

Navigate to voyant-tools.org. There are several options for uploading text, including typing or pasting text directly into the text box, uploading one or more files, or using an existing sample corpus. Use the "Upload" button to select the appropriate text files. This text will form the corpus, a representation of the collection of text documents and their metadata.

Selecting the Appropriate Functions to Analyze Text

The text analysis is presented in an interactive interface, with the following functionality:

- Cirrus, a word cloud tool that includes a slider function to add or reduce the number of terms included in the cloud. Cirrus provides functionality to remove stop words such as *the*, terms that are frequently used but irrelevant. The "Scale" function allows users to view a word cloud for an entire corpus, as well as for individual documents. The "Export" function enables users to export the word cloud as a PNG (portable network graphic) or SVG (scalable vector graphics) file.
- Terms, a term frequency matrix with search syntax capability, will count how many times a term occurs and can be used to identify high-frequency terms.
- Links, a network visualization of collocated terms, those that occur in proximity, can help identify compound words or key phrases.
- Reader, to view all the text in a corpus.
- Terms Berry, which identifies high-frequency terms and their collocates.
- Trends, a line graph of term distributions.
- Document Terms, a table view of term frequencies for individual documents.
- Summary, an overview of the corpus.
- Documents, a table of the documents in the corpus with functions for modifying them.
- Phrases, which shows repeating word sequences organized either by frequency of repetition or number of words in each repeated phrase.
- Contexts, which shows individual keyword occurrences and their surrounding text.
- Bubblelines, which visualizes the frequency and distributions of terms.
- Correlations, which explores terms whose frequencies rise and fall together.

To determine how terms are used differently across databases, analyze and compare results from the three database files. Multiple files can be uploaded to Voyant Tools in a single session, where their results can be compared. This combined approach will also allow for examination of the entire corpus, providing a comprehensive analysis and summation of results. The Voyant Tools Help guide, https://voyant-tools.org/docs/#!/guide, offers additional information, including tips for getting started with Voyant Tools, general information, and an overview of available functions.[22]

R and RStudio

Voyant Tools is powerful and capable of quickly analyzing text to optimize search term selection by revealing frequently used terms, creating interactive visualizations that illustrate term distribution and correlation, and indicating occurrences of repeating word sequences. However, its functionality is limited. Users have restricted control over how data are presented, which can be limiting when preparing visualizations to share with research teams or reviewers or in publications. In addition, the ability to analyze specific elements of a corpus is restricted in Voyant Tools. Librarians who begin to experience growing pains and crave more control over text analysis functionality should consider R for text analysis.

R is a free statistical programming language that offers robust text analysis options and includes capabilities to transform and filter text.

R is a free statistical programming language that offers robust text analysis options and includes capabilities to transform and filter text. R allows for customizable tailoring of corpus elements. For librarians new to R and programming, analyzing text with R can be an ideal and practical introduction. There are several free resources for learning R, including the "Programming with R" guide from the Software Carpentry Foundation, a popular option among librarians and information specialists.[23]

Readers who followed the preceding steps of this tutorial developed search strategies and used Voyant Tools to execute basic analysis on text data from a development set. R provides deeper analysis options. Users can also use R to write scripts, which can automate tasks so they can be executed in future projects.

Setting Up an R Project Work Space

Download the R software, available at r-project.org. Also download RStudio, which allows for interactive execution of R functionality. A free desktop download is available at rstudio.com.

Create a working directory and a new project. In RStudio, use the File menu option to "Create a New Project." Select the "New Directory" option. Choose "New Project" when prompted for the "Project Type." Name your directory. For this example, we have used the name *SRDataset*. Select the "Create Project" button and proceed.

Use the toolbar option (shown in Figure 1) to create a new R script. Save the new script using the File > Save As option. A script will help track actions and will also be useful for re-creating analyses by simply updating the text files.

Creating a Data Set

Import text data from the development set. For an abstract analysis, export the title and abstracts into a text file. For full-text analysis, copy and paste the entire text of the articles into a text file, which should include a file for each of the selected databases. These text files will serve as the data set for the analysis. Note that these files will already exist if the preceding steps of this tutorial were followed. Upload your text files into a new R directory folder.

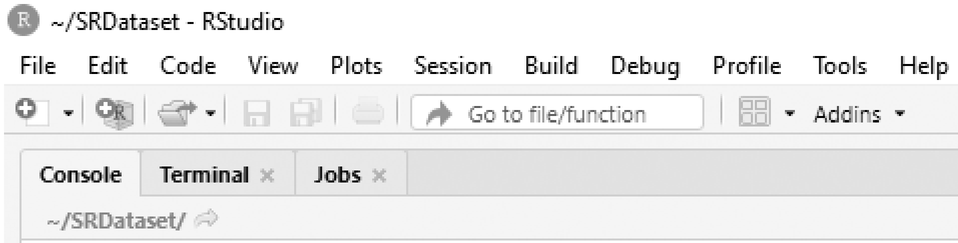

Figure 1. The RStudio toolbar. To open a new empty script, click on the "File" icon in the upper left corner of the toolbar. It looks like a white square with a white plus sign in a green circle. Clicking the icon opens a new file menu. Click the "R Script" menu option, and the script editor will open with an empty script.

For this example, the folder will be named *SRDataset*. Assume three research databases were searched—PubMed, CINAHL (Cumulative Index to Nursing and Allied Health Literature), and Web of Science. The working R directory will contain three plain-text files with the file names *pubmed.txt*, *cinahl.txt*, and *wos.txt*.

Loading the Necessary Packages

Load the text mining package called *tm*, used for text mining, and the word cloud package called *wordcloud*, to create visualizations. An interactive version of this portion of the tutorial is included in the supplemental files as an R Shiny markdown document. This interactive document can be executed using the "Run Document" command in R Studio and will allow the user to see and test a few functions.

The appropriate packages will need to be installed and attached to the R Studio work space. If using the interactive R Shiny markdown file, load the "rmarkdown" and "shiny" packages. In the Console pane, type the following commands to install the necessary packages:

```
install.packages("tm")
install.packages("wordcloud")
install.packages("rmarkdown")
install.packages("shiny").
```

It will take a few minutes for the packages and their dependencies to load. After a successful download is complete, a message with the location of the packages will appear.

Attaching the Packages to the Work Space

In the Console, type the following commands to attach the packages to your work space:

```
library(tm)
library(wordcloud)
library(rmarkdown)
library(shiny).
```

Creating a Corpus

A corpus provides the structure for text analysis. We will take a comprehensive approach to analyze the corpus, merging the three individual text files. Use the “c” function to combine the three text files into a single list. Call this combined list *filelist*:

filelist<-c(“pubmed.txt”, “cinahl.txt”, “wos.txt”).

Processing the Corpus

The *filelist* command joins the text files, but the formatting makes it difficult to read and distinguish the text. Separate each file onto a new line using the “readLines()” function, which reads lines of text from an input file. The “lapply()” function will apply a function to each element of the list. Use both functions to create a separated list called *text*:

text<-lapply(filelist, FUN=readLines).

In programming, an argument is a value passed to a routine or function. The “collapse” argument will collapse each vector element into a single character element, which allows text from each file to be presented on a single line. Name this combined, readable list *corpus*:

corpus<-lapply(text, FUN=paste, collapse=“ “).

Once executed, new data and values will appear in the Environment pane. Figure 2 illustrates the Environment pane with the newly created values for *corpus* and *filelist*, along with their values.

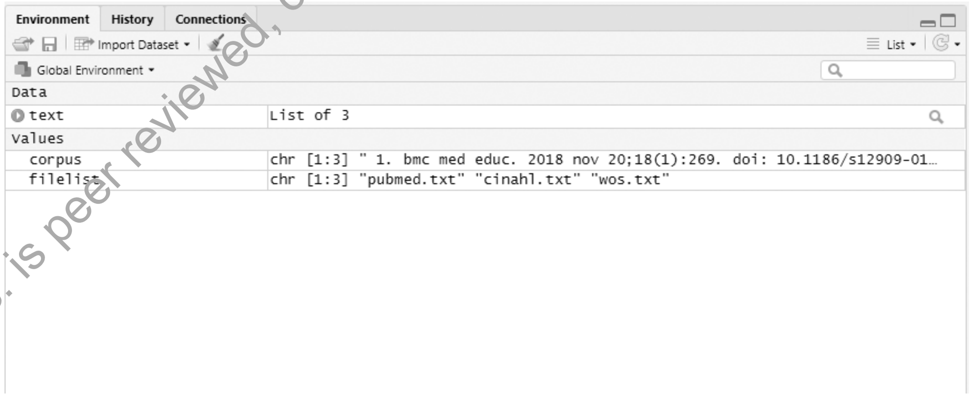

Figure 2. The RStudio Environment pane displays all data and values that have been entered into the Console pane.

Cleaning the Corpus

Making Characters Lowercase

Cleaning the corpus will help eliminate irrelevant clutter. For this example, cleaning will consist of removing stop words—commonly used but irrelevant words such as *the*—and will also include eliminating unnecessary white space, punctuation, numbers, and capitalization.

R is case-sensitive, and capitalization could skew the results of the text analysis. Making all characters lowercase will force lowercased and previously uppercased words to be treated equally. For example, *the* will be handled the same as *The*. Use the "tolower()" function to make characters lowercased:

```
corpus <-tolower(corpus).
```

Removing Punctuation

The "gsub" function will replace all matches of a string. Use "gsub" to remove punctuation marks:

```
corpus <-gsub('[[:punct:]]', '', corpus).
```

Removing Stop Words

Stop words are frequently occurring words that do not have significant meaning. The "removeWords" function will eliminate them, helping to generate more relevant results. See the "stop word" function page in the R documentation guide at https://www.rdocumentation.org/ for an overview of English-language stop words:

```
corpus2 <-removeWords (corpus, stopwords('english')).
```

Removing White Space

The "stripWhitespace" function will remove unwanted white space from the beginning and end of strings:

```
corpus3 <-stripWhitespace(corpus2).
```

Removing Numbers

The "removeNumbers" function will eliminate numbers from a corpus:

```
corpus4 <-removeNumbers(corpus3).
```

Viewing the Cleaned Corpus

Use the "view()" function to see the cleaned corpus:

```
View(corpus4).
```

With a cleaned corpus, we can begin the text analysis.

Identifying High-Frequency Terms

The "wordcloud()" function is useful for identifying high-frequency words across a corpus and for illustrating less common words that could be relevant to the search strategy. R provides the functionality to control many aspects of word cloud visualizations. For example, the random.order=FALSE argument will arrange terms in decreasing frequency, with high-frequency terms in the middle of the cloud and lesser-used words or phrases on the outer edges. The "col" argument will sort terms hierarchically and assign colors based on frequency:

wordcloud(corpus4, random.order=FALSE, col=rainbow(3)).

Using the "Help" Feature

When writing code, warnings and errors might be encountered. Warnings occur when R can fully execute a command but encounters one or more problems. Errors happen when R has a situation that forces it to stop before it can execute the command. The "Help" feature provides useful information for avoiding and resolving warnings and errors and also includes detailed information for using packages, their functions, and their arguments. Figure 3 illustrates how the "Help" feature provides information on the range of arguments available to customize the "wordcloud()" function.

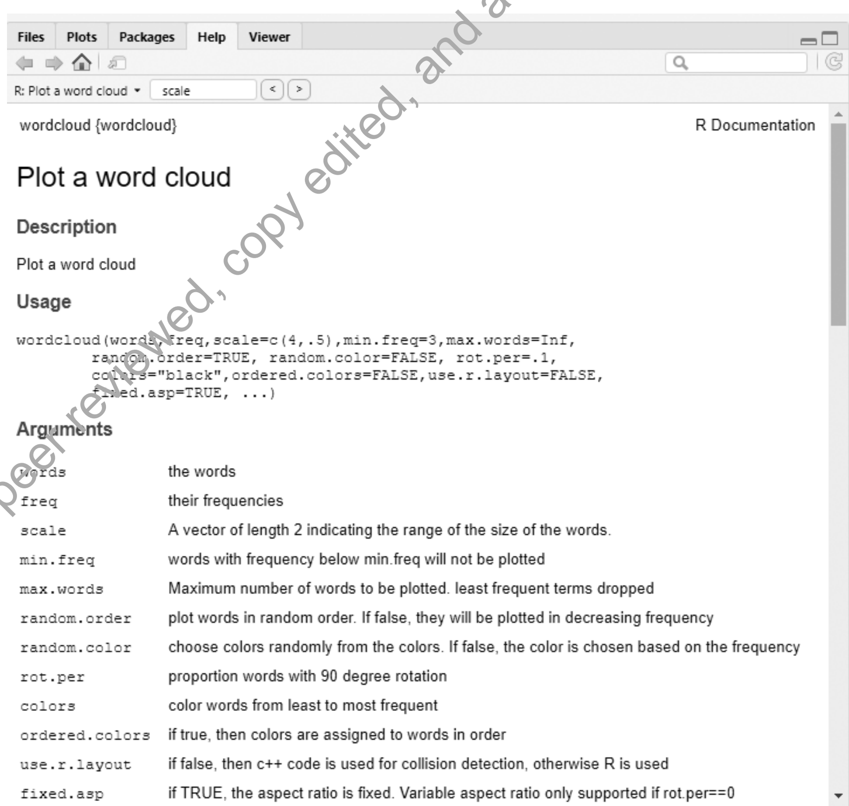

Figure 3. The RStudio "Help" feature provides detailed information for using packages, their functions, and their arguments—that is, values passed between functions. This screen shows the range of arguments available to customize the "wordcloud()" function.

Creating a Term-Document Matrix

A term-document matrix can be used to illustrate term frequency. Use the "VectorSource()" function to create a corpus for the R tm package to read. Then, create a term-document matrix:

```
corpus5 <-VCorpus(VectorSource(corpus4))
termdocmatrix <- TermDocumentMatrix(corpus5).
```

Use the "colnames" function to rename the matrix columns from their default numerical values to something more meaningful. First, name the matrix *mat*:

```
mat <- as.matrix(termdocmatrix).
```

Use the "colnames()" function to show the current, default numerical column names:

```
colnames (mat).
```

To rename the columns, assume column 1 is text from PubMed, column 2 is from Web of Science, and column 3 is from CINAHL:

```
colnames(mat) <- c("PMed", "WoS", "CNHL").
```

Assessing Term Use across Databases with a Comparison Cloud

A comparison cloud, "comparison.cloud()," will illustrate the results of each of the three text files as a single visualization. This can reveal how terms are used, similarly or differently, across databases. Use the "scale" argument to adjust the visualization size to include the desired number of terms:

```
comparison.cloud(mat, scale=c(3.0, 0.3)).
```

Requesting Terms Used a Certain Number of Times

Apply the term frequency function, "findFreqTerms()," to view terms used a specified number of times. Identify words or phrases that occur 10 or more times:

```
findFreqTerms(termdocmatrix, 10).
```

Identifying Correlating Terms

The "findAssocs()" function will reveal correlations between specific terms. For example, suppose the term *sports* is a major theme in the research question and we want to know which other terms are associated with the word, with at least a 0.99 correlation across the corpus:

```
findAssocs(termdocmatrix, "hackathon", 0.99).
```

Discussion

Text mining tools help librarians improve search precision, increase search sensitivity, and translate search strategies across multiple research databases. Entry-level text mining tools, like Voyant Tools, offer text analysis capabilities with minimal investment in time and resources. Though R and RStudio have a steeper learning curve, their combined functionality allows for greater control over a corpus, and R scripts allow functions to be easily reproduced and reused for future projects.

When combined with an intuitive search strategy development approach, text mining tools help reduce biases by improving the objectivity, transparency, and reproducibility of search strategies.

When combined with an intuitive search strategy development approach, text mining tools help reduce biases by improving the objectivity, transparency, and reproducibility of search strategies. The use of the Hausner team's four-step objective search strategy process, or a similar procedure, further increases the transparency and reproducibility of search strategy development.

Next Steps

For more information about the functions referenced in this tutorial and for additional information about text mining applications, see the following R packages: tm,[24] textmineR,[25] revtools,[26] and litsearchr.[27] In addition to supporting the identification of search terms, text mining tools can be used to underpin other systematic review tasks, such as citation screening[28] and prioritizing references.[29] These functions may be useful for librarians engaged in review roles beyond information retrieval. For more information about the automation of systematic review service in academic libraries, see the "Systematic Review Automation LibGuide" from the University of North Carolina at Chapel Hill Health Sciences Library: https://guides.lib.unc.edu/automation.[30]

Conclusion

Text mining tools, when combined with an intuitive search development process, help reduce biases in search term selection. With its low barrier of entry, interactive interface, and ability to execute useful text analysis functions, Voyant Tools provides a practical introduction for librarians seeking to use text mining tools to inform search term generation. However, Voyant Tool's preprogrammed functionality is limited. Librarians who want more flexible data analysis will find that R supports the most robust options for text analysis and visualization. Though R has a steeper learning curve than Voyant Tools, R scripts allow tasks to be quickly and easily reproduced for future use.

Bethany S. McGowan is an assistant professor of information studies and a health sciences information specialist in the Library of Engineering and Science at Purdue University Libraries and School of Information Studies in West Lafayette, Indiana; she may be reached by email at: bmcgowa@purdue.edu.

Notes

1. Miranda Cumpston, Tianjing Li, Matthew J. Page, Jacqueline Chandler, Vivian A. Welch, Julian P. T. Higgins, and James Thomas, "Updated Guidance for Trusted Systematic Reviews: A New Edition of the Cochrane Handbook for Systematic Reviews of Interventions," *Cochrane Database of Systematic Reviews* 10, 10 (2019): ED000142, https://doi.org/10.1002/14651858.ED000142; Julian P. T. Higgins and Sally Green, eds., *Cochrane Handbook for Systematic Reviews of Interventions: Cochrane Book Series* (Hoboken, NJ: John Wiley, 2008).
2. Carol Lefebvre, Julie Glanville, Simon Briscoe, Anne Littlewood, Chris Marshall, Maria-Inti Metzendorf, Anna Noel-Storr, et al., "Searching for and Selecting Studies," chap. 4 in Higgins and Green, *Cochrane Handbook for Systematic Reviews of Interventions*.
3. Melissa L. Rethlefsen, Ann M. Farrell, Leah C. Osterhaus Trzasko, and Tara J. Brigham, "Librarian Co-Authors Correlated with Higher Quality Reported Search Strategies in General Internal Medicine Systematic Reviews," *Journal of Clinical Epidemiology* 68, 6 (2015): 617–26.
4. Rohit Borah, Andrew W. Brown, Patrice L. Capers, and Kathryn A. Kaiser, "Analysis of the Time and Workers Needed to Conduct Systematic Reviews of Medical Interventions Using Data from the PROSPERO [International Prospective Register of Systematic Reviews] Registry," *BMJ Open* 7, 2 (2017): e012545.
5. Mark Petticrew, "Systematic Reviews from Astronomy to Zoology: Myths and Misconceptions," *BMJ* 322, 7278 (2001): 98–101; Julia H. Littell, "Systematic Reviews in the Social Sciences: A Review," *Evidence & Policy: A Journal of Research* 2, 4 (2006): 535–37; Richard Andrews, "The Place of Systematic Reviews in Education Research," *British Journal of Educational Studies* 53, 4 (2005): 399–416; Maura Borrego, Margaret J. Foster, and Jeffrey E. Froyd, "Systematic Literature Reviews in Engineering Education and Other Developing Interdisciplinary Fields," *Journal of Engineering Education* 103, 1 (2014): 45–76.
6. Amy Riegelman and Megan Kocher, "A Model for Developing and Implementing a Systematic Review Service for Disciplines outside of the Health Sciences," *Reference & User Services Quarterly* 58, 1 (2018): 22–27; Elizabeth Moreton, "Systematic Review Automation: Learn about SR [systematic review] Automation," Health Sciences Library, University of North Carolina University Libraries, 2020, https://guides.lib.unc.edu/automation/learn.
7. Carole Lunny, Joanne E. McKenzie, and Steve McDonald, "Retrieval of Overviews of Systematic Reviews in MEDLINE Was Improved by the Development of an Objectively Derived and Validated Search Strategy," *Journal of Clinical Epidemiology* 74 (2016): 107–18.
8. Elke Hausner, Charlotte Guddat, Tatjana Hermanns, Ulrike Lampert, and Siw Waffenschmidt, "Development of Search Strategies for Systematic Reviews: Validation Showed the Noninferiority of the Objective Approach," *Journal of Clinical Epidemiology* 68, 2 (2015): 191–99; Lunny, McKenzie, and McDonald, "Retrieval of Overviews of Systematic Reviews in MEDLINE Was Improved by the Development of an Objectively Derived and Validated Search Strategy"; Eliza Grames, Andrew Stillman, Morgan W. Tingley, and Chris S. Elphick, "An Automated Approach to Identifying Search Terms for Systematic Reviews Using Keyword Co-Occurrence Networks," *Methods in Ecology and Evolution* 10, 10 (2019): 1645–54.
9. Claire Stansfield, Alison O'Mara-Eves, and James Thomas, "Text Mining for Search Term Development in Systematic Reviewing: A Discussion of Some Methods and Challenges," *Research Synthesis Methods* 8, 3 (2017): 355.
10. Rethlefsen, Farrell, Osterhaus Trzasko, and Brigham, "Librarian Co-Authors Correlated with Higher Quality Reported Search Strategies in General Internal Medicine Systematic Reviews"; Stansfield, O'Mara-Eves, and Thomas, "Text Mining for Search Term Development in Systematic Reviewing," 355–65.
11. Stansfield, O'Mara-Eves, and Thomas, "Text Mining for Search Term Development in Systematic Reviewing."

12. Christopher W. Belter, "Citation Analysis as a Literature Search Method for Systematic Reviews," *Journal of the Association for Information Science and Technology* 67, 11 (2016): 2766–77; James Thomas, John McNaught, and Sophia Ananiadou, "Applications of Text Mining within Systematic Reviews," *Research Synthesis Methods* 2, 1 (2011): 1–14; Grames, Stillman, Tingley, and Elphick, "An Automated Approach to Identifying Search Terms for Systematic Reviews Using Keyword Co-Occurrence Networks"; Stansfield, O'Mara-Eves, Thomas, "Text Mining for Search Term Development in Systematic Reviewing"; Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou, "Using Text Mining for Study Identification in Systematic Reviews: A Systematic Review of Current Approaches," *Systematic Reviews* 4, 5 (2015); Hausner, Guddat, Hermanns, Lampert, and Waffenschmidt, "Development of Search Strategies for Systematic Reviews"; Lunny, McKenzie, and McDonald, "Retrieval of Overviews of Systematic Reviews in MEDLINE Was Improved by the Development of an Objectively Derived and Validated Search Strategy"; Yindalon Aphinyanaphongs, Ioannis Tsamardinos, Alexander Statnikov, Douglas Hardin, and Constantin F. Aliferis, "Text Categorization Models for High-Quality Article Retrieval in Internal Medicine," *Journal of the American Medical Informatics Association* 12, 2 (2005): 207–16.
13. Stansfield, O'Mara-Eves, and Thomas, "Text Mining for Search Term Development in Systematic Reviewing."
14. Ibid.
15. Angela J. Spencer and Jonathan D. Eldredge, "Roles for Librarians in Systematic Reviews: A Scoping Review," *Journal of the Medical Library Association* 106, 1 (2018): 46–56.
16. Thomas, McNaught, and Ananiadou, "Applications of Text Mining within Systematic Reviews"; O'Mara-Eves, Thomas, McNaught, Miwa, and Ananiadou, "Using Text Mining for Study Identification in Systematic Reviews"; Stansfield, O'Mara-Eves, and Thomas, "Text Mining for Search Term Development in Systematic Reviewing."
17. Elke Hausner, Siw Waffenschmidt, Thomas Kaiser, and Michael Simon, "Routine Development of Objectively Derived Search Strategies," *Systematic Reviews* 1, 1 (2012): 1.
18. Ibid., 19.
19. Ibid.
20. Madelynn Dickerson, "A Gentle Introduction to Text Analysis with Voyant Tools," University of California Irvine Humanities Commons, 2017, http://www.escholarship.org/uc/item/6jz712sf.
21. Megan E. Welsh, "Review of Voyant Tools," *Collaborative Librarianship* 6, 2 (2014): 96–98.
22. Stéfan Sinclair and Geoffrey Rockwell, "Voyant Tools Help," Voyant Tools, 2020, https://voyant-tools.org/docs/#!/guide.
23. Software Carpentry Foundation, "Programming with R," https://swcarpentry.github.io/r-novice-inflammation/.
24. Ingo Feinerer, "Introduction to the tm (text mining) Package Text Mining in R," Comprehensive R Archive Network, 2019, https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf.
25. Tommy Jones and William Doane, "TextmineR: Functions for Text Mining and Topic Modeling," Version 3.0.4, Comprehensive R Archive Network, 2018, https://cran.r-project.org/web/packages/textmineR/index.html.
26. Martin J. Westgate, "Revtools: An R Package to Support Article Screening for Evidence Synthesis," *Research Synthesis Methods* 10, 4 (2019): 606–14.
27. Grames, Stillman, Tingley, and Elphick, "An Automated Approach to Identifying Search Terms for Systematic Reviews Using Keyword Co-Occurrence Networks."
28. Alexandra Bannach-Brown, Piotr Przybyla, James Thomas, Andrew S. C. Rice, Sophia Ananiadou, Jing Liao, and Malcolm Robert Macleod, "Machine Learning Algorithms for Systematic Review: Reducing Workload in a Preclinical Review of Animal Studies and Reducing Human Screening Error," *Systematic Reviews* 8, 1 (2019): 1–12; Babatunde Kazeem Olorisade, Pearl Brereton, and Peter Andras, "Reproducibility of Studies on Text Mining for

Citation Screening in Systematic Reviews: Evaluation and Checklist," *Journal of Biomedical Informatics* 73 (2017): 1–13; Gerald Gartlehner, Gernot Wagner, Linda Lux, Lisa Affengruber, Andreea Dobrescu, Angela Kaminski-Hartenthaler, and Meera Viswanathan, "Assessing the Accuracy of Machine-Assisted Abstract Screening with DistillerAI: A User Study," *Systematic Reviews* 8, 1 (2019): 277.

29. Austin J. Brockmeier, Meizhi Ju, Piotr Przybyla, and Sophia Ananiadou, "Improving Reference Prioritisation with PICO [patient/population, intervention, comparator, outcomes] Recognition," *BMC Medical Informatics and Decision Making* 19, 1 (2019): 256.
30. Moreton, "Systematic Review Automation."