A Gentle Introduction to Topic Modeling Using Python

by Micah D. Saxton

Abstract

Topic modeling is a data mining method used to understand and categorize large corpora of data. As such, it is a tool that theological librarians can use in their professional workflows and scholarly practices. In this article, I provide a gentle introduction to topic modeling using the Python programming language for those who have no prior knowledge of the topic. I begin with a conceptual overview of topic modeling which does not rely on the complicated mathematics behind the process. Then, I illustrate topic modeling by providing a step-by-step example of building a topic model using *Theological Librarianship* as an example corpus. This example ends with an analysis of the success of the model and suggestions for improvement. Finally, I comment on the practical application of topic modeling for library workflows.

Introduction

As information professionals, theological librarians benefit from an awareness of tools that have been developed to help sort through and understand the increasing deluge of information we encounter every day. A topic model is one such tool that, when employed correctly, can allow its user to understand large corpora of documents. These documents can be *anything*: scholarly articles, historical resources, or tweets. The only requirement is that they are available in a digital format.

There is an unofficial tradition in introductory literature about programming or technical computer topics: the tradition of providing a "gentle introduction." The idea behind a gentle introduction is to invite the reader into a topic without getting overly technical. It is in that spirit that I offer the following gentle introduction to topic modeling. The introduction will be gentle because I will not discuss the mathematics behind topic modeling nor will I provide the Python code I wrote as a part of this project. For those interested in the code, I have made it available on Github.[1]

Conceptual Overview

In general, a model is a representation of an object that captures what is important about that object in a way that is more easily used. For example, an architectural model scales down a building so that it can be more readily observed without sacrificing the proportions and detail of the original. A topic model is similar; it models a corpus of documents, not by scaling them down, but by generating topics that are representative of the content of the document corpus. To be more specific, a topic model is a probabilistic model used to discover topics, or latent structures, across a collection of documents.[2] These topics can then be used to organize the documents or develop controlled vocabularies that describe the documents.

While the algorithms that lie behind a topic model are complex, a conceptual explanation of how a topic model operates is possible.[3] An underlying assumption is that any given document contains within it latent topics.[4] These topics are latent because they are not explicitly signaled by the author. Topics, in turn, are composed of words with similar semantic

[1] See https://github.com/msaxton/tl_topic_model.

[2] Rubayyi Alghamdi and Khalid Alfalqi, "A Survey of Topic Modeling in Text Mining," *International Journal of Advanced Computer Science and Applications* 6, no. 1 (2015): 147–153. https://doi.org/10.14569/IJACSA.2015.060121.

[3] For the math behind the algorithm used here see David M. Blei et al., "Latent Dirichlet Allocation," *Journal of Machine Learning Research* 3, no. 4/5 (May 15, 2003): 993–1022.

[4] Megan R. Brett, "Topic Modeling: A Basic Introduction," *Journal of Digital Humanities*, April 8, 2013, http://journalofdigitalhumanities.org/2-1/topic-modeling-a-basic-introduction-by-megan-r-brett/.

Micah D. Saxton is Collections Development and Reference Librarian at the Ira J. Taylor Library at Iliff School of Theology in Denver.

domains. To put all of this another way, words can be grouped together to form topics, and topics are grouped together to form documents. The probability of a topic being present in a document can be measured by the words used in the document.

Take the front page of a newspaper as an example: a newspaper contains a number of articles (documents) which can be taken together as a corpus. There are a number of words that may appear in several different articles such as "Middle East," "law," "border," "immigration officer," "senate," "visa," "illegal," "policy," "undocumented," "asylum," "court," "president," "legislation," and "international." A topic model algorithm would iterate through these articles and record patterns of word co-occurrence. On the basis of such patterns it may sort "Middle East," "border," "immigration officer," "visa," illegal," "undocumented," and "asylum" into one group and sort "policy," "senate," "law," "court," "president," "legislation," and "international" into another. Each of these groupings are considered topics and we could label the first "immigration" and the second "politics."

Of course, some words could be placed in both groupings. The topic model algorithm would also assign a probability value of each term belonging to the group to which it has been assigned. Additionally, the topic model algorithm would assign a probability of each grouping being found in a given document. It is important to remember that this process occurs on the basis of any number of algorithms and *not* on the basis of human conceptual categories, even though the latter may be used to interpret the former. Insofar as a topic model does not sort documents on the basis of pre-defined categories, it is an instance of "unsupervised machine learning."[5]

Examples of Application

There are a number of useful applications of topic modeling for librarians and scholars. One such application is to use a topic model to assist in the classification of documents. The manual classifying of documents poses a number of challenges. There is the practical challenge of how long it takes for human catalogers to classify documents. For example, the efforts of librarians at the National Medical Library who are responsible for assigning medical subject headings (MeSH) to articles published in biomedical fields take an extraordinary amount of time and are expensive.[6] MeSh contains over 26,000 terms, and indexers look through an entire article in order to assign those subject headings. According to Ramakanth Kavuluru and Yuan Lu, this process could be streamlined by supplementing some of the process with unsupervised machine learning techniques such as a topic model.[7] These authors argue that topic models could be constructed on the corpus of already indexed biomedical articles which could then be used to classify incoming articles.[8] This process still requires humans to map topics to controlled vocabularies, but such efforts would go a long way toward streamlining the process.

The challenges to manual classification of documents are not just practical; there are theoretical challenges as well. Scientists are often in the business of producing new knowledge, but that new knowledge may not always relate neatly to current subject headings. Arho Suominen and Hannes Toivanen argue specifically, "The central novelty of unsupervised-learning methods in classifying scientific knowledge is that they virtually eliminate the need to fit new-to-the-world knowledge into known-to-the-world definitions."[9] By "unsupervised-learning methods" these

[5] Graham Upton and Ian Cook, "Machine Learning," in *A Dictionary of Statistics*, 3rd ed. (Oxford University Press, 2014), http://www.oxfordreference.com.du.idm.oclc.org/view/10.1093/acref/9780199679188.001.0001/acref-9780199679188-e-2380.

[6] Ramakanth Kavuluru and Yuan Lu, "Leveraging Output Term Co-Occurrence Frequencies and Latent Associations in Predicting Medical Subject Headings," *Data & Knowledge Engineering*, Special issue following the 18th International Conference on Applications of Natural Language Processing to Information Systems (NLDB'13), 94 (November 1, 2014): 189–201. https://doi.org/10.1016/j.datak.2014.09.002.

[7] Kavuluru and Lu, "Leveraging Output."

[8] Kavuluru and Lu, "Leveraging Output."

[9] Arho Suominen and Hannes Toivanen, "Map of Science with Topic Modeling: Comparison of Unsupervised Learning and Human-Assigned Subject Classification," *Journal of the Association for Information Science & Technology* 67, no. 10 (October 2016): 2465, https://doi.org/10.1002/asi.23596.

authors mean topic modeling. They go on to argue for the use of topic modeling as a way of classifying documents by comparing topic modeling with human-assigned subject classification.[10] These authors do not suggest that humans turn over the entirety of document classification to machines, but they do suggest that topic modeling has certain advantages when it comes to classifying new knowledge.

These are not the only examples of information professionals using topic modeling to aid in the classification of documents. Topic modeling has been used in the fields of public health to organize information about substance abuse and depression among teens.[11] It has been used to automatically tag webpages.[12] It has also been used to organize documents for more efficient information retrieval.[13] In short, topic modeling is a valuable tool for information professionals who organize information. There is no reason why theological librarians cannot also use topic modeling to their own advantage. The first step is to learn how topic models work, and the second step is to think creatively about how topic models can be applied to the professional activities and scholarly pursuits of theological librarians.

Step-by-Step Example of Building a Topic Model

In what follows I will illustrate the process of topic modeling by providing an example of all the major steps from gathering documents to initializing a topic model. For this example, I used *Theological Librarianship* (*TL*) as my corpus to be modeled. *TL* is a relatively small corpus (at the time of writing there are under 350 articles) so it does not illustrate the full potential of topic modeling, which can be effective on corpora with documents numbering in the thousands. But it provides the unique challenge of being a generally heterogeneous corpus focused on a niche discipline. This makes a topic model more difficult because the topics need to have a greater degree of nuance.

Software

There are a number of software packages available in different programming languages for generating topic models. MALLET is one popular option; it is written in Java, but there are wrappers available in both Python and R.[14] For those who are more familiar with the programming language R, there are a few different packages available such as topicmodels[15] and lda.[16] For those with a preference for the Python programming language, Genism is an increasingly popular topic modeling package.[17] Genism is notable for its scalability (it can handle corpora containing tens of thousands of documents) and its user interface. Any of these software options are sufficient for doing a topic model, but here I have selected Genism primarily because of the popularity and the ease of use of the Python language. That said, many of the steps in what follows are described on a conceptual level which will be of use no matter what software one uses.

---

[10] Suominen and Toivanen, "Map of Science with Topic Modeling."

[11] Wang Shi-Heng et al., "Text Mining for Identifying Topics in the Literatures about Adolescent Substance Use and Depression," *BMC Public Health* 16 (2016), https://doi.org/10.1186/s12889-016-2932-1.

[12] Maria Lin and David W. Cheung, "An Automatic Approach for Tagging Web Services Using Machine Learning Techniques," *Web Intelligence* (2405-6456) 14, no. 2 (April 2016): 99–118. https://doi.org/10.3233/WEB-160334.

[13] Shoaib Jameel, Wai Lam, and Lidong Bing, "Supervised Topic Models with Word Order Structure for Document Classification and Retrieval Learning," *Information Retrieval Journal* 18, no. 4 (August 2015): 283–330. https://doi.org/10.1007/s10791-015-9254-2.

[14] For an introduction to MALLET see Shawn Graham, Scott Weingart, and Ian Milligan, "Getting Started with Topic Modeling and MALLET," *Programming Historian*, September 2, 2012, https://programminghistorian.org/lessons/topic-modeling-and-mallet.

[15] Bettina Grun and Kurt Hornik, "Topicmodels: An R Package for Fitting Topic Models," *Journal of Statistical Software* 40, no. 13 (May 1, 2011), https://doaj.org.

[16] Jonathan Chang, *Lda: Collapsed Gibbs Sampling Methods for Topic Models*, version 1.4.2, n.d., https://cran.r-project.org/web/packages/lda/lda.pdf.

[17] R. Rehurek and P. Sojka, "Software Framework for Topic Modeling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta: ELRA, 2010).

Genism is the tool that does the computational work of constructing the model; all it needs is a properly prepared corpus of documents. Gensim operates with three core concepts: corpus, vector, and model.[18] The *corpus* is the collection of documents from which the model is generated. These documents must be prepared in a specific way to be of use to Gensim; namely, each document of the corpus must be made into a list of words. Each word in a corpus can be thought of as a feature of the corpus. If every word in the corpus is a feature, it is clear that each document will have many, but not all, features available in the corpus. A *vector*, then, is a representation of each document that tallies each of the features that document contains. Consider two similar sentences:

(a) Corpora are composed of many documents.

(b) Documents contain many features.

If these two sentences are taken together as a small corpus, we could say that the corpus has eight unique features: "corpora," "are," "composed," "of," "many," "documents," "contain," and "features." We can then represent the sentences in this way:

	corpora	are	composed	of	many	documents	contain	features
(a)	1	1	1	1	1	1	0	0
(b)	0	0	0	0	1	1	1	1

The rows in the above table (excluding the header) are vector representations of each sentence which tally how many times each feature appears in each sentence. Finally, in the context of Gensim, a *model* is a representation of a corpus. It is a way of referring to the transformation of one document representation to another. As a basic example, the table above is a model of sentences a and b. Sentences a and b have been transformed from sentences to a table.

Step-by-Step Procedure

The actual process of topic modeling involves five major steps: (1) create a document corpus, (2) preprocess the text to gather the most informative features, (3) process the text into a corpus that can be used by Gensim (that is, turn each document into a vector), (4) build the topic model, and (5) analyze the topic model.

Create a Document Corpus

Creating a document corpus is more than just making a list of the documents you want to model. You must get a digital text into a form that the computer program can read. This means that you must have a plain text string. It is also helpful at this first stage to collect relevant metadata for each document for later reference. When I created a document corpus for this example, I did a few things. First, with the permission of the editorial board of *Theological Librarianship*, I wrote a Python script (a) to scrap metadata off the website which included author, title, issue date, and a few other relevant links, and (b) to download a pdf of each article. (I deliberately left out the full issue pdfs as well as the "About the Journal" section at the beginning of each issue because the full issue pdfs were redundant for this model and the "About the Journal" section was outside of the content I wanted to model.) Second, I used a Python package called PyPDF2 to extract the plain text out of the pdf. This text extraction was not perfect. Often where the pdf had a "th" as in "theology," the program read it as a "~" and displayed "~eology"; other times the program read a "th" as "º" and displayed "ºeology." This error occurred with many other instances of "th" as well as other combinations like "fi" and "ff". These kinds of data errors are common when dealing with text data and must be accounted for when preparing a text corpus.

[18] R. Rehurek, "Gensim: Topic Modelling for Humans," accessed February 4, 2018, https://radimrehurek.com/gensim/intro.html.

Preprocess the Text

There is a cliché in data science that is true for creating topic models: "Garbage in, garbage out." Preprocessing is the process of "cleaning up" the text as best as one can, so that the model can be built on the most informative features. I would not want the program to count "˜eology" and "ºeology" as two different word types because they are both misspellings of "theology." To normalize the text, I replaced "˜" and "º" with "th." There were a number of other substitutions that had to be made as well.[19] But pre-processing does not only need to normalize misspelled words, it also needs to change all the letters to lower case so that "Theology" and "theology" are not counted as separate word types. Additionally, uninformative words like "a," "the," "in," "because," and "and" need to be removed. These kinds of words are called "stop-words" and are eliminated from a topic model because they are too common to be informative. Preprocessing also requires that the text be lemmatized. Lemmatization is the process of reducing each word to its lexical form. I did not want the program to count "reduce," reducing," and "reduced" as three separate words; rather, I want to lemmatize each to "reduce." Finally, after errors have been accounted for, each word lowercased, stop words removed, and the remaining words lemmatized, the last part of preprocessing is to tokenize each document. Tokenization is the process of breaking the plain text into individual units on the level of words. (Actually, one could tokenize a plain text down to sentences as well, but that would not be useful for a topic model.) Let us take a random sentence from *Theological Librarianship* as an example of preprocessing.

The unprocessed plain text looks like this:

> "˜ousands of volumes were contributed to the library by retiring pastors and seminary professors."

Account for errors from text extraction:

> "thousands of volumes were contributed to the library by retiring pastors and seminary professors."

Remove stop words:

> "thousands volumes contributed library retiring pastors seminary professors."

Lemmatize:

> "thousand volume contribute library retire pastor seminary professor."

Tokenize:

> ["thousand", "volume", "contribute", "library", "retire", "pastor", "seminary", "professor"]

After preprocessing, each document in the corpus has become a list of word tokens which can be processed by Gensim into a topic model.

Process the Text into a Gensim Corpus

Much of the difficult human work comes in the preprocessing phase. Processing the corpus into something Gensim can model is simpler (on the human side of things; the computer does most of the work here). The first thing Gensim does here is to create what it calls a "dictionary" based on the corpus given to it. The primary function of the dictionary is to assign an integer ID to each unique word type in the corpus and map that integer to the word itself. The reason for this is that computers can work faster with numbers than they can with words. When Gensim builds the topic model, it uses the integer ID rather than the word for faster processing. That said, an important decision needs to be made before Gensim builds the dictionary. There is an option to eliminate words that may not be informative features. Words that are used too frequently are not informative because they fail to distinguish one document from another. On the other hand, words that are used too rarely are not informative because there is not enough commonality to group similar documents together. For this topic model, I eliminated any word that is used in more than 50 percent of the entire corpus and any word that is used in less than five documents in the entire corpus.

[19] For specifics see https://github.com/msaxton/tl_topic_model/blob/master/tl_topic_model.py.

After making the dictionary, Gensim can now create a corpus to train the topic model. The corpus created here is different from the starting corpus in that now each document is represented by a series of number pairs. The first number in the pair represents the integer ID assigned to a word type and the second number in the pair represents how often that word type occurs. A document in this corpus may then looks something like: [(9,5) (72, 1) (56, 2)…] where in the first pairing "9" may refer to the word "library" (for example) and "5" refers to the number of times the word is used in the document.

Initialize the Topic Model

After Gensim makes the dictionary and creates the corpus, initializing the model is not complicated for the human (but it is very complicated for the computer). Topic models use different algorithms to work through the text corpus and find patterns of word co-occurrence. Gensim provides a few different algorithms, but for this topic model I selected the widely used Latent Dirichlet Allocation (LDA).[20] In this step there is an important parameter that must be considered: how many topics does one wants to find in the corpus? This decision is not automated by the program and must be decided upon by the human user. Too few or too many topics will render a model which is not informative. For this project I ran an experiment and created four models: one with twenty-five topics, one with fifty, one with seventy-five, and one with one hundred. After looking at the results, I decided that the model with fifty topics is the most informative.

Analyze the Topic Model

How do we know if the topic model was successful? The program itself does not know the meanings of the words, it just groups the words on the basis of patterns of co-occurrence. It is up to the human user to evaluate if the model is helpful or not.

For our analysis, let us first see what a topic looks like:

(39,
'0.023*"space" + 0.023*"building" + 0.020*"project" + 0.014*"room" +
'0.013*"college" + 0.009*"area" + 0.009*"plan" + 0.009*"service" +
'0.008*"house" + 0.008*"build"')

This topic is simply labeled "39." The program does not name the topics, it just generates them. We could label this topic something like "library arrangement" or "physical structures." The list of ten words are the ten most prominent words in the topic. The number associated with each word indicates its probability of belonging to the topic. Conceptually, these ten words fit into a coherent semantic domain. The model can also indicate which documents are associated with each topic. In order to narrow the association between topics and documents in an informative way, I set the threshold at 30 percent, meaning that for a document to be associated with a topic, it must have a probability score of at least 0.3. In the case of topic 39, the model associated the following *Theological Librarianship* articles with this topic:

- Rebekah Hall, "The James E. Rolfing Memorial Library, Trinity International University," *Theological Librarianship* 5, no. 1 (2012): 9-11.
- Andrew G. Kadel, "The Christoph Keller, Jr. Library, General Theological Seminary," *Theological Librarianship* 5, no. 1 (2012): 12-15.
- Joshua Michael, "Murphey Memorial Library, Baptist College and Seminary," *Theological Librarianship* 5, no. 1 (2012): 16-18.
- Eileen K. Saner, "Library of the Associated Mennonite Biblical Seminary," *Theological Librarianship* 5, no. 1 (2012): 19-22.
- Garrett Trott, "The Corban University Library," *Theological Librarianship* 5, no. 1 (2012): 23-25.
- Audrey Williams, "The John Richard Allison Library, Regent College," *Theological Librarianship* 5, no. 1 (2012): 26-29.

[20] Blei et al., "Latent Dirichlet Allocation."

Each of these articles comes from volume 5 issue 1 from a special forum titled "The Reshaping of Libraries." In this instance, the topic model successfully identified a meaningful topic and properly associated relevant articles with the topic.

Topic 0 provides us another example (note: computers generally start a list with 0 not 1):

(0,
'0.045*"church" + 0.021*"pope" + 0.013*"christian" + 0.012*"fie" +
0.011*"john" + 0.010*"franci" + 0.009*"middle" + 0.008*"francis" + '
0.008*"catechism" + 0.007*"life"')

There are clearly some problems with words in this topic. The word "franci" should be "francis" and the word "fie" is likely the result of poor text extraction from the pdf. Nonetheless, this group of words clearly represents a coherent semantic domain. The topic model associated the following articles with this topic with a 30 percent or higher probability:

- Keith Edward Lemna, "Pope Francis' Strong Thought," *Theological Librarianship* 7, no. 2 (2014): 45-53.
- Lorraine H. Olley, "Benedict Biscop: Benedictine, Builder, Bibliophile," *Theological Librarianship* 7, no. 1 (2014): 30-37.
- Michael R. Mitchell, "Christian Catechetical Texts," *Theological Librarianship* 5, no. 2 (2012): 92-94.
- Matthew Baker, "Christian Traditions in the Contemporary Middle East," *Theological Librarianship* 4, no. 1 (2011): 68-74.
- Fred Guyette, "The Literature of Ecclesiology: A Ten Year Retrospective," *Theological Librarianship* 4, no. 1 (2011): 75-90.
- Katharina Penner, "Information Behaviour(s) of Theologians: A Literature Review," *Theological Librarianship* 2, no. 1 (2009): 67-82.
- Fred Guyette, "An Open Access Source for the Study of Religion and the Law: The Proceedings of the Old Bailey: London's Central Criminal Court 1674-1913," *Theological Librarianship* 1, no. 2 (2008): 28-37.

While not all of these articles have an explicit focus on Pope Francis or Catholicism, they each are related to the topics of church or ecclesiology in ways that other articles in *TL* are not.

Topic 36 appears to be primarily about Methodism, but "consortium" and a few other terms form this topic, so there is room for improvement here:

(36,
'0.054*"methodist" + 0.030*"church" + 0.024*"united" + 0.023*"methodism" + 0.018*"consortium" +
0.014*"denomination" + 0.011*"culture" + 0.011*"episcopal" + 0.009*"draw" + 0.009*"century"')

The topic model associated the following articles with this topic:

- Christopher J. Anderson, "We Desire Everything Illustrating the History of Methodism That We Can Procure: Examining the Methodist Collections at Drew University," *Theological Librarianship* 6, no. 1 (2013): 9-15.
- James Wiser, "'Playing Well With Others:' New Opportunities for Library Consortia," *Theological Librarianship* 5, no. 2 (2012): 43-47.
- Mark R. Teasdale, "Growth of Declension: Methodist Historians' Treatment of the Relationship Between the Methodist Episcopal Church and the Culture of the United States," *Theological Librarianship* 3, no. 2 (2010): 34-44.

The three topics above provided examples of coherent semantic domains (even though there is room for improvement in each). However, not all topics generated by this model are as informative.

Topic 7 is an example of a "junk" topic that lacks informative cohesion:

> (7,
> '0.020*"religion" + 0.013*"dictionary" + 0.013*"religious" + 0.013*"entry" + 0.009*"theology" + 0.008*"subject" + 0.008*"essay" + 0.008*"editor" + 0.008*"cover" + 0.007*"index"')

There is nothing wrong with these words per se, but they do not form a coherent semantic domain. It is no surprise then, that the topic model associated 40 different articles with this topic having a probability score of 0.3. The only seeming connection between these articles seems to be that they are about librarianship, books, or theological books generally. Such an insight may be helpful about a multi-disciplinary corpus, but is hardly helpful in a corpus of articles about theological librarianship.

The final topic for this analysis is topic 29:

> (29,
> '0.022*"site" + 0.016*"bible" + 0.016*"search" + 0.015*"http" + 0.012*"user" + 0.011*"available" + 0.011*"version" + 0.010*"www" + 0.010*"link" + 0.009*"digital"')

This topic forms a coherent semantic domain. We could label this topic something like "online resources" or "digital objects." However, the model identified nearly thirty articles with this topic having a probability score of 0.3 or higher. While it may be the case that these articles all address online resources, it may also be the case that the letter sets "http" and "www" are too general, perhaps coming from citations, to be informative features.

Based on this brief examination of some of the topics generated and the articles associated with those topics can we say that this model was successful? Does it model the corpus of *Theological Librarianship* in helpful ways? I would say that it was moderately successful with much room for improvement. The good news is that improvements may be possible.

If I were tasked with classifying the articles in the *Theological Librarianship* with a topic model with more precision, I would make the following improvements: (1) The plain text extracted from the pdfs needs more nuanced cleaning. A mechanism to distinguish when "~" was inserted for a "th" from when it was inserted for a "fi" needs to be developed. Alternatively, the plain raw texts could be extracted from the epub files using a Python package such as eBookLib, but that method will have its own issues to address. Improving on the quality of the raw plain text would go a long way toward initiating a more useful topic model. (2) It may be useful to identify bi-grams in the text and include those as significant features. Gensim, as well as a few other Python packages such as the Natural Language Toolkit (NLTK) provide this kind of functionality. (3) The parameters that are intended to eliminate less informative features could perhaps be adjusted. All of this work would be worth the effort because once a sufficient model is initiated, it could not only be used to categorize the existing articles in *Theological Librarianship*, it could also be used to categorize new articles as they are published.

Conclusion

By way of conclusion we can ask: How can the process of topic modeling be applied to library related projects? Alternatively, why should a librarian bother to learn this skill? Insofar as many library workflows aim at the organization of information, there is potential for topic modeling to enhance the activities of a librarian. Here are some examples.

First, many academic libraries serve as the repository for theses written by graduate students. Using a topic model trained on an existing collection of theses would allow a librarian to automate much of the process of cataloging new theses. I am not suggesting that the entire process of cataloging be automated, but cataloging workflows for unpublished materials like graduate theses could be made more efficient by mapping a topic model's topics to a controlled vocabulary used by an institution. This is especially true if a library has a sizable collection of graduate theses in digital form.

Second, and similarly, as digital archival collections grow, there is a growing need to make these collections more discoverable. An archivist could train a topic model on an existing digital collection and use that model to group sub-collections and assign key words to digital objects. The digital archivist could then use the topic model as the basis for suggesting "related items" to any digital object.

Finally, many libraries collect qualitative survey data. If that data becomes too large to be easily manageable, a topic model could help librarians break it down into categories that are easy to process. These are just a few examples of how librarians can apply topic models to their work. These examples are by no means exhaustive, and with a little effort the readers of this journal could think of many more. In fact, the more that librarians familiarize themselves with skills like topic modeling, the more creative librarians can get in applying such skill sets.

The goal of this gentle introduction was to expose the readers of *Theological Librarianship* to topic models as a way to understand and process large corpora of documents. Toward that end, I discussed topic modeling on a conceptual level, I pointed out a few applications, and then I provided a step-by-step example of building a topic model of *Theological Librarianship* including an analysis of that example. That topic model was somewhat successful in modeling *Theological Librarianship* in helpful ways, but I also made a few suggestions whereby that model may be improved. My hope is that this article inspires a curiosity in its readers to learn more about topic modeling and find creative applications of this process for their professional workflows and scholarly practices.

Bibliography

Blei, David M., Andrew Y. Ng, Michael I. Jordan, and John Lafferty. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3, no. 4/5 (May 15, 2003): 993–1022.

Brett, Megan R. "Topic Modeling: A Basic Introduction." *Journal of Digital Humanities*, April 8, 2013. http://journalofdigitalhumanities.org/2-1/topic-modeling-a-basic-introduction-by-megan-r-brett/.

Chang, Jonathan. *Lda: Collapsed Gibbs Sampling Methods for Topic Models* (version 1.4.2), n.d. https://cran.r-project.org/web/packages/lda/lda.pdf.

Graham, Shawn, Scott Weingart, and Ian Milligan. "Getting Started with Topic Modeling and MALLET." *Programming Historian*, September 2, 2012. https://programminghistorian.org/lessons/topic-modeling-and-mallet.

Grun, Bettina, and Kurt Hornik. "Topicmodels: An R Package for Fitting Topic Models." *Journal of Statistical Software* 40, no. 13 (May 1, 2011). https://doaj.org.

Jameel, Shoaib, Wai Lam, and Lidong Bing. "Supervised Topic Models with Word Order Structure for Document Classification and Retrieval Learning." *Information Retrieval Journal* 18, no. 4 (August 2015): 283–330. https://doi.org/10.1007/s10791-015-9254-2.

Kavuluru, Ramakanth, and Yuan Lu. "Leveraging Output Term Co-Occurrence Frequencies and Latent Associations in Predicting Medical Subject Headings." *Data & Knowledge Engineering*, Special issue following the 18th International Conference on Applications of Natural Language Processing to Information Systems (NLDB'13), 94 (November 1, 2014): 189–201. https://doi.org/10.1016/j.datak.2014.09.002.

Lin, Maria, and David W. Cheung. "An Automatic Approach for Tagging Web Services Using Machine Learning Techniques." *Web Intelligence* (2405-6456) 14, no. 2 (April 2016): 99–118. https://doi.org/10.3233/WEB-160334.

Rehurek, R. "Gensim: Topic Modelling for Humans." Accessed February 4, 2018. https://radimrehurek.com/gensim/intro.html.

Rehurek, R., and P. Sojka. "Software Framework for Topic Modeling with Large Corpora." In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010.

Rubayyi Alghamdi, and Khalid Alfalqi. "A Survey of Topic Modeling in Text Mining." *International Journal of Advanced Computer Science and Applications* 6, no. 1 (2015): 147–153. https://doi.org/10.14569/IJACSA.2015.060121.

Shi-Heng, Wang, Yijun Ding, Weizhong Zhao, Yung-Hsiang Huang, Roger Perkins, Wen Zou, and James Chen. "Text Mining for Identifying Topics in the Literatures about Adolescent Substance Use and Depression." *BMC Public Health* 16 (2016). https://doi.org/10.1186/s12889-016-2932-1.

Suominen, Arho, and Hannes Toivanen. "Map of Science with Topic Modeling: Comparison of Unsupervised Learning and Human-Assigned Subject Classification." *Journal of the Association for Information Science & Technology* 67, no. 10 (October 2016): 2464–76. https://doi.org/10.1002/asi.23596.

Upton, Graham, and Ian Cook. "Machine Learning." In *A Dictionary of Statistics*. Oxford University Press, 2014. http://www.oxfordreference.com.du.idm.oclc.org/view/10.1093/acref/9780199679188.001.0001/acref-9780199679188-e-2380.