ARTICLE

Using Machine Learning and Natural Language Processing to Analyze Library Chat Reference Transcripts

Yongming Wang

ABSTRACT

The use of artificial intelligence and machine learning has rapidly become a standard technology across all industries and businesses for gaining insight and predicting the future. In recent years, the library community has begun looking at ways to improve library services by applying AI and machine learning techniques to library data. Chat reference in libraries generates a large amount of data in the form of transcripts. This study uses machine learning and natural language processing methods to analyze one academic library's chat transcripts over a period of eight years. The built machine learning model tries to classify chat questions into a category of reference or nonreference questions. The purpose is to predict the category of future questions by the model with the hope that incoming questions can be channeled to appropriate library departments or staff.

INTRODUCTION

Since the beginning of this century, artificial intelligence (AI) and machine learning (ML) have been used in almost all industries and businesses to gain knowledge and insights and predict the future. The large amount of data available has helped to accelerate the application of AI and ML in stunning speed. To follow this technology trend, the library community has begun looking at ways to improve library services by applying AI and ML techniques to library data.

Stanford University Library is one of the pioneers in the research and application of ML and AI in the library. The mission of its Library AI Initiatives states: "The Library AI initiative is a program to identify, design, and enact applications of artificial intelligence that will help us make our rich collections more easily discoverable, accessible, and analyzable."[1] In 2019, Stanford University Library hosted the second International Conference on AI for Libraries, Archives, and Museums, titled Fantastic Future.[2]

Many academic libraries have implemented chat reference services as a way to support student learning and academic research on campus. Chat reference serves as an important channel to connect the library's resources and services to the campus community.[3] Over the years, libraries have accumulated a large amount of data in the form of chat transcripts. Analyzing the content of transcripts can help the library understand users' information needs, deploy library human resources more efficiently, and improve the quality of the chat reference service.

The College of New Jersey is a midsize academic library that serves a campus with 7,000 college students, most of them undergraduates. The library began to use Springshare's LibChat in 2014. The chat service is freely accessible online from the library's website, and anyone can initiate a chat by asking an initial question through the chat box. Approximately 8,000 chat transactions have been accumulated over the past eight years.

Yongming Wang (wangyo@tcnj.edu) is Systems Librarian, The College of New Jersey. © 2022.

This study aims to use machine learning and natural language processing (NLP) techniques to build a classification model to categorize all available questions into two categories: reference and nonreference. By doing so, we hope that the model can automatically classify future chat questions received into either the reference question category or the nonreference question category, and channel the question to the appropriate library department or staff.

LITERATURE REVIEW

Traditionally, the analysis of chat transcripts has used qualitative or simple quantitative methods (e.g., chat frequency, duration). To better understand chat service quality and patrons' information needs, librarians must manually review and read through chat transcripts, which requires a lot of time and effort.[4] In recent years, however, the library field has started to witness the application of AI and ML techniques to analyze library data, including chat transcripts, in order to quickly and efficiently gain more insight into user information needs and information seeking patterns.

Megan Ozeran and Piper Martin used topic modeling, a ML method, to analyze library chat reference conversations. The purpose of their project was to identify the most popular topics asked by library patrons in order to improve the chat reference service and to train the library staff.[5]

The Brigham Young University library implemented a machine learning–based tool to perform various text analysis on transcripts of chat reference to gauge patron satisfaction levels and to classify patrons' questions into several categories.[6] Jeremy Walker and Jason Coleman used ML and NLP techniques to build models that predict the relative difficulty of incoming chat reference questions. They tested their large sample size of chat transcripts on hundreds of models. Their aim was to help library professionals and management improve chat reference services in the library.[7]Another ML topic modeling project of was carried out by HyunSeung Koh and Mark Fienup. Their study applied pLSA (Probabilistic Latent Semantic Analysis) to library chat data over a period of four years, resulting in more accurate and interpretable topics and subjects compared with results by human qualitative evaluation.[8]

Another interesting ML project on chat reference data was conducted by Ellie Kohler. This project used a machine learning model to analyze chat transcripts for sentiment and topic extraction.[9]

In addition to library chat data, ML has been also used to analyze other library data, including library digital collections and library tweet data. Jeremiah Flannery applied NLP summarization techniques on a special library digital collection of Catholic pamphlets. This project tried to automatically generate a summary for each digitized pamphlet by using NLP's BERT Extractive technique and Gensim python package.[10] Sultan M. Al-Daihani and Alan Abrahams conducted a text mining analysis of academic libraries' tweets. They used a tool called PamTAT developed by the Pamplin College of Business at Virginia Polytechnic Institute and State University. Pamplin is a Microsoft Excel–based interface to the NLP NLTK package written in Python. The purpose of their analysis was to try to identify the most common topics or subject keywords of the tweets by 10 large academic libraries. In addition, they also ran Harvard General Inquirer for semantic and sentiment analysis of the tweets.[11]

Other applications of ML techniques in the academic library include analyzing library operations such as acquisition. In 2019, Kevin W. Walker and Zhehan Jiang from the University of Alabama used a machine learning method called adaptive boosting (AdaBoost) to predict demand-driven acquisition (DDA).[12] Carlos G. Figuerola, Franciso Javier Carcia Marco, and Maria Pinto used the topic modeling technique, specifically the Latent Dirichlet Allocation, to identify the main topics and categories of the 92,705 publications in the domain of library and information science from 1978 to 2014.[13]

PAIR (Projects in Artificial Intelligence Registry) is a repository and online global directory of AI projects in higher education. It is maintained by the University of Oklahoma Libraries. The aim of PAIR is to foster cross-institutional collaboration and to support grant activity in the field of artificial intelligence and machine learning in higher education.[14]

Public libraries have started to seriously look at the application and impact of AI in the library. Frisco Public Library in Texas has developed a series of applications and programs to help train library staff in AI. They also developed artificial intelligence maker kits, including Google AIY Voice Kit, for circulation. They even provide introductory Python lessons to the public.[15]

BACKGROUND OF NLP AND ML

Natural language processing is a multidiscipline field that involves linguistics, computer science, and machine learning. By using computer algorithms, NLP tries to build a machine learning model that is applied to large amounts of data in order to make predictions or decisions. The data in NLP is natural language data, that is, data in plain and unstructured textual form in any language.

There are many types of applications of NLP and ML in business or people's daily lives. Especially with the popularity of Internet, there is a tremendous increase in and accumulation of textual data, such as social media networks and customer online chat services. Major applications of NLP include sentiment analysis on social media data, topic modeling in digital humanities, text classification, speech recognition, search box auto correct, and auto completion, etc. The use cases are countless.

In general, there are two types of ML: supervised learning vs. unsupervised learning. In supervised learning, the dataset fed to the model is labelled in advance to classify data or predict outcomes accurately whereas unsupervised learning is a type of algorithm that learns patterns from unlabeled or untagged data.

No matter which type, all ML and NLP techniques involve a series of general steps in any project, also called the ML/NLP pipeline.

1. Data collection, which involves obtaining the raw textual data and usually means downloading data from some remote server or service.
2. Data preprocessing, which is necessary for any project, large or small, because the raw textual data is unstructured data and is not ready to be fed to the model for computing processes. Data preprocessing usually includes removing punctuations, changing all letters to lowercase, tokenization, removing stop words, and stemming or lemmatization.
3. Feature engineering, which is optional but often very useful.
4. Text vectorization, which is the final step before feeding the data to the model. The purpose is to transform the text into some kind of value in numbers.

5. Model building, evaluation, and optimization, which involves multiple cycles until the optimal or desired results are achieved.
6. Implementation, which is the final step in implementing the model to the real world.

METHODOLOGY

For this ML/NLP project, the raw data came from the chat transcripts repository downloaded from Springshare's server. From 2014 to 2021, a total of 8,000 chat reference transactions were logged. These transactions formed the raw dataset for model building and testing with this project.

Because of the nature of the data, i.e., textual data, Python was chosen for this project. And the two major Python packages used in the project are NLTK and scikit-learn. NLTK (Natural Language Toolkit) is a suite of libraries and programs for natural language processing for English language. NLTK supports classification, tokenization, and stemming, tagging, parsing, and semantic reasoning functionalities. Scikit-learn is a Python module built on NumPy, SciPy, and Matplotlib. Featuring various classification, regression, and clustering algorithms, including support-vector machines, random forest, gradient boosting, k-means and DBSCAN, scikit-learn is a simple and efficient tool for predictive data analysis. Scikit-learn is one of the most popular Python modules for any ML project.

Data Collection
Data collection includes both data gathering and data preparation. Data gathering is the process of downloading the 8,000 initial questions into an Excel file. Data preparation deals with the initial data clean up, such as removing the blank rows. The most important task of data preparation is data labelling. Because this is a supervised-learning ML project, all questions must be labeled as either reference question (label=Yes) or nonreference question (label=No) by hand. Then, all labeled questions (dataset) are fed to the ML model for either training purposes or testing purposes. See table 1 for an example of data after the preparation step.

Table 1. Sample questions with Yes or No labels

Question sequential number	Label	Question
3979	Yes	Working on an Alumni Reunion presentation. I need to know …
3980	Yes	would a book with this call number: DS559.8.D7 G68 1991 …
3981	No	Would a Rutgers student be able to take out a textbook from …
3982	Yes	Would I be able to find mathematics textbooks by Pearson on …
3983	No	Would I be able to log in to find an article if I am an alumni of …
3984	Yes	Would it be possible to help me find a online essay?
3985	No	Would like to renew: Huguenots [videorecording] / music by …
3986	Yes	would like to request for a course description catalog from Fall …
3987	No	Would someone be able to ask room 414 to quiet down please?
3988	No	would someone be able to come up to floor 3 and tell people to …

Data Preprocessing

Data preprocessing is the first programming step in the pipeline of this ML/NLP project. Data preprocessing transforms the raw data into a more digestible form so that the ML model can perform better and achieve the desired results. One of the purposes of data preprocessing is to remove insignificant and nonmeaningful words such as "a," "the," "and," etc., as well as punctuation, from the textual data. Removing nonmeaningful and stop words from the corpus allows for a better result in the ML model, allowing it to deal only with significant and meaningful words.

It is also necessary to apply lowercase formatting to all letters. While we as humans know that lowercase and uppercase words have the same meaning, the computer will treat them as having different meanings. For example, "cat" and "Cat" are two different words to the computer. Tokenization involves splitting the sentences into a list of individual words by removing spaces between words. The last step of data preprocessing is stemming or lemmatizing, which is to find the semantic meaning of a group of related words. In other words, this process explicitly correlates words with similar meanings. For instance, run, running, runner will become "run"; library and libraries will become "librari"; goose and geese will become "goose."

Feature engineering involves creating a new feature, or transforming the current feature. The purpose of feature engineering is to help the model make better predictions. This step is optional but often very helpful if done right. In this project, a new feature called "question length" was created. "Question length" was based on the assumption that the average length of reference questions is longer than the average length of nonreference questions. If such is the case, the ML model will benefit by using this new feature to make better decisions. Figure 1 is a histogram of question length distribution. The distribution of reference questions is represented in blue; nonreference questions are represented in yellow.

Figure 2 shows a sample result list following completion of data preprocessing and feature engineering. From left to right, it lists the result after each step. The question length feature (Question_len column) must follow the original question because this step is based on the original question before any other steps. The Question_lemma column is the result after all preprocessing steps.
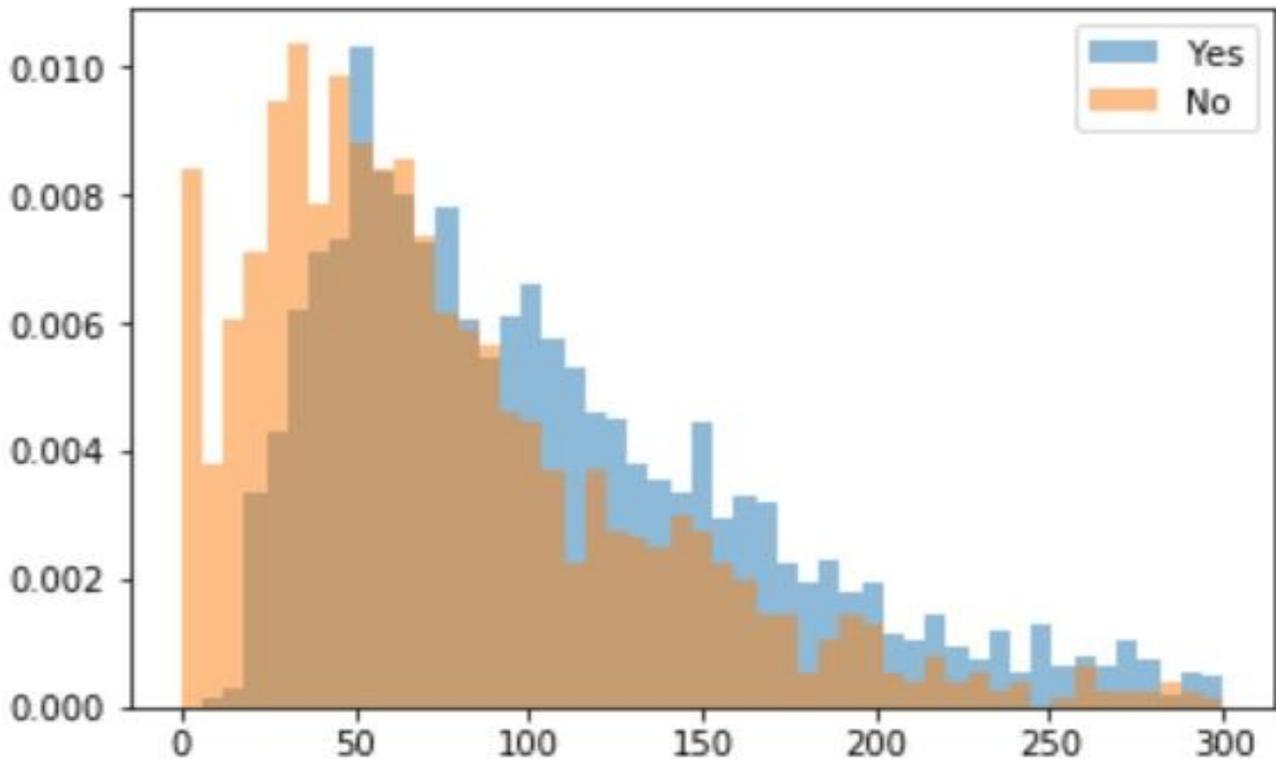
Figure 1. Histogram of question length distribution.

Figure 2. Results from data preprocessing and feature engineering.

Text Vectorization
The purpose of text vectorization is to transform the text data into numeric data so that the ML algorithms and Python can understand and use that data to build a model. The basic idea is to build an n-dimensional vector of numerical features that present some object. The three most popular text vectorizations are count vectorization, n-grams vectorization, and TF-IDF vectorization. TF-IDF stands for term frequency–inverse document frequency. Because it is weighted, it is more accurate. Figure 3 shows the result of TF-IDF vectorization.

	1900	1946	2	2285	319	3rd	4019a2	4025a5r5	4th	8pm11pm	...	undergo	underlin	understand	us	use	wan'
0	0.00000	0.00000	0.00000	0.00000	1.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
1	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.29434	0.000000	0.000000	0.000000	0.2943·
2	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
3	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
4	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.144628	0.00000	0.144628	0.000000	0.000000	0.0000(
5	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.232018	0.0000(
6	0.26305	0.26305	0.00000	0.26305	0.0	0.000000	0.26305	0.26305	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
7	0.0000	0.00000	0.57735	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.288675	0.000000	0.0000(
8	0.0000	0.00000	0.00000	0.00000	0.0	0.691340	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
9	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
10	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.319615	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
11	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
12	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.691340	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
13	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.5	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
14	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(
15	0.00000	0.00000	0.00000	0.00000	0.0	0.000000	0.00000	0.00000	0.000000	0.0	...	0.000000	0.00000	0.000000	0.000000	0.000000	0.0000(

Figure 3. Result of TF-IDF vectorization.

Model Building, Testing, and Evaluation
The first step of model building is to divide the dataset into two sets, one for a training model and one for model testing. Normally we use 80% of the data for training and 20% for testing. After feeding the training data to the model, we feed the testing data as new data to the model to predict the Yes or No label based on the pattern that the model builds through the training data. Testing data were initially labeled by humans and are 100 percent accurate. By comparing the labels predicted by the model with the labels in the training data, we would know how the model performs, and make changes, if necessary, to the model parameters.

Scikit-learn contains several ML models. This project used two popular models: random forest and gradient boosting. The random forest model builds many decision trees and computes them at the same time. Then the final decision is made by majority vote. Because it computes at the same time, it is more efficient and fast. The gradient boosting model builds one tree at a time. Each new tree helps correct errors made by previously trained trees, and then the model is boosted (optimization) by reward or penalty. In theory, gradient boosting should yield better results than random forest. Nevertheless, it is slower and consumes more resources.

The confusion matrix was used to evaluate the performance of the two models. There are three parameters in the confusion matrix. They are accuracy, precision, and recall. Accuracy equals true positive plus true negative and then divided by the total. Precision equals true positive divided by

true positive plus false positive. Recall equals true positive divided by true positive plus false negative.

Usually there is a tradeoff between precision and recall. Recall shows the number of false negatives or the percentage of false negatives of the total, and precision shows the number of false positives. False negative means that the model predicts the reference question as a nonreference question. False positive means that the model predicts the nonreference question as a reference question. Which is more important for the model to catch, false positive or false negative? The answer depends on the actual situation. In our case, false negative is more serious than false positive because we did not want the real reference questions to be predicted as nonreference questions. However, it was acceptable if the nonreference questions were predicted as reference questions. Therefore, we wanted the least amount of false negatives, which meant the largest recall value possible.

RESULTS AND ANALYSIS

Table 2 lists the result from both models.

Table 2. Results of random forest model and gradient boosting model

	Precision	Recall	Accuracy	Fit time	Predict time
Random forest model	0.914	0.964	0.912	2.489 s	0.15 s
Gradient boosting model	0.904	0.948	0.894	97.786 s	0.064 s

In general, any parameter values above 0.9 (90%) are very good. Looking at and comparing those results, we can see that both models performed well. Nevertheless, the random forest model had better results than that of the gradient boosting model in all three parameters. In addition, the fit time of the random forest model was much shorter than that of the gradient boosting model. Even though the predict time of the random forest model is slightly longer than that of the gradient boosting model, it is relatively insignificant.

Therefore, the random forest model was chosen for the final model for this project.

CONCLUSION AND FUTURE WORK

In this pilot study, we used the classification modeling of NLP and ML techniques to divide patrons' chat questions into two categories: reference questions and nonreference questions. The purpose of the model is to predict the category of future questions received through chat so that library staff and professionals can provide faster, more efficient reference services. Two machine learning models were tested: random forest and gradient boost. After comparing results from each model, it was concluded that the random forest model showed better results.

What is the next step after the model is built? A potential use of this model is to implement it as a plugin or feature enhancement for the online chat application. The model can function as the filter to direct incoming questions to either reference librarians if the question is predicted as a reference question by the model, or to library staff or graduate student assistants if the question is predicted by the model as a nonreference question. This will be especially useful for libraries that have busy online chat service transactions.

Further work can be done to make the model be multicategory. For example, a multicategory model can go beyond two categories and include categories for information seeking, citation help, printing help, noise complaints, interlibrary loan questions, spams, etc. Thus, the model can send the question to the relevant department or library personnel accordingly.

ENDNOTES

[1] "Stanford University Library AI Initiative," Stanford University Library, https://library.stanford.edu/projects/artificial-intelligence.

[2] "Fantastic Futures: 2nd International Conference on AI for Libraries, Archives, and Museums," (2019), Stanford University Library, https://library.stanford.edu/projects/fantastic-futures.

[3] Christina M. Desai and Stephanie J. Graves, "Cyberspace or Face-to-Face: The Teachable Moment and Changing Reference Mediums," *Reference & User Services Quarterly* 47, no. 3 (Spring 2008): 242–55, https://www.jstor.org/stable/20864890.

[4] Sharon Q. Yang and Heather A. Dalal, "Delivering Virtual Reference Services on the Web: An Investigation into the Current Practice by Academic Libraries," *Journal of Academic Librarianship* 41, no. 1 (November 2015): 68–86, https://doi.org/10.1016/j.acalib.2014.10.003.

[5] Megan Ozeran and Piper Martin, "Good Night, Good Day, Good Luck: Applying Topic Modeling to Chat Reference Transcripts," *Information Technology and Libraries* 38, no. 2 (June 2019): 49–57, https://doi.org/10.6017/ital.v38i2.10921.

[6] Christopher Brousseau, Justin Johnson, and Curtis Thacker, "Machine Learning Based Chat Analysis," *Code4Lib Journal,* no. 50 (2021), https://journal.code4lib.org/articles/15660.

[7] Jeremy Walker and Jason Coleman, "Using Machine Learning to Predict Chat Difficulty," *College & Research Libraries* 82, no. 5 (2021), https://doi.org/10.5860/crl.82.5.683.

[8] HyunSeung Koh and Mark Fienup, "Topic Modeling as a Tool for Analyzing Library Chat Transcripts," *Information Technology and Libraries* 40, no. 3 (2021), https://doi.org/10.6017/ital.v40i3.13333.

[9] Ellie Kohler, "What Do Your Library Chats Say? How to Analyze Webchat Transcripts for Sentiment and Topic Extraction" (17th Annual Brick & Click Libraries Conference, Maryville, Missouri: Northwest Missouri State University, 2017).

[10] Jeremiah Flannery, "Using NLP to Generate MARC Summary Fields for Notre Dame's Catholic Pamphlets," *International Journal of Librarianship* 5, no.1 (2020): 20–35, https://doi.org/10.23974/ijol.2020.vol5.1.158.

[11] Sultan M. Al-Daihani and Alan Abrahams, "A Text Mining Analysis of Academic Libraries' Tweets," *The Journal of Academic Librarianship* 42, no. 2 (2016): 135–43, https://doi.org/10.1016/j.acalib.2015.12.014.

[12] Kevin W. Walker and Zhehan Jiang, "Application of Adaptive Boosting (AdaBoost) in Demand-Driven Acquisition (DDA) Prediction: A Machine-Learning Approach," *The Journal of Academic Librarianship* 45, no. 3 (2019): 203–12, https://doi.org/10.1016/j.acalib.2019.02.013.

[13] Carlos G. Figuerola, Francisco Javier Garcia Marco, and Maria Pinto, "Mapping the Evolution of Library and Information Science (1978–2014) Using Topic Modeling on LISA," *Scientometrics* 112 (2017): 1507–35, https://doi.org/10.1007/s11192-017-2432-9.

[14] "Projects in Artificial Intelligence Registry (PAIR): A Registry for AI Projects in Higher Ed," University of Oklahoma Libraries, https://pair.libraries.ou.edu/.

[15] Thomas Finley, "The Democratization of Artificial Intelligence: One Library's Approach," *Information Technology and Libraries* 38, no. 1 (2019): 8–13, https://doi.org/10.6017/ital.v38i1.10974.