Topic Model Documentation

Title: Identifying Multiscale Basin Management Challenges and Current Research Priorities based on Topic Modeling of the Mississippi River Basin

Author(s): Joshua "Jay" Wimhurst[1,2], Jennifer Koch[3], Renee McPherson[1,2]

Affiliations: *[1]South Central Climate Adaptation Science Center, [2]University of Oklahoma Department of Geography and Environmental Sustainability, [3]Wageningen University (The Netherlands) Laboratory of Geo-Information Science and Remote Sensing*

Lead Model Developer: Joshua "Jay" Wimhurst

Synopsis: Since the year 1990, over 2,000 scientific publications, government/non-profit reports, books, book chapters, and pamphlets about the Mississippi River Basin (MRB) have been made publicly available on the internet. These documents comprise a wide range of different topics pertaining to the uses, challenges, and critical functions of the MRB on a range of spatiotemporal scales. Given the importance of the MRB for sustaining vulnerable habitats, goods trade, and climate regulation, synthesizing this literature to better understand the past research priorities of the MRB would better inform these priorities into the future. Identifying these priorities in the form of research topics reveals common and unique socio-environmental challenges facing the MRB, challenges that may exist within and across states and sub-basins, or that were priorities only in certain decades. We present here a Latent Dirichlet Allocation (LDA) model that identifies the most commonly studied topics in the MRB literature, with the literature on which the LDA algorithm is trained being customizable by spatial (basin-wide, sub-basin, state) and temporal (1990s, 2000s, 2010s, 2020s) scale. The topics produced by the LDA algorithm reveal spatiotemporally common and unique research priorities from the last 35 years, and consequently inform the research directions that should be taken next. Informing future research directions is particularly important given the context of the pressure that climate change places on the MRB's ongoing and future management.

Software Requirements: Python 3.12 or newer, with all relevant modules installed.

Model Availability: The following can be downloaded via GitHub or Open Science Framework:

- One empty folder for holding all saved model outputs ("Model Training Results.zip") and another folder containing templates for word cloud shapes ("Word Cloud Templates.zip"). *These must be decompressed before use.*
- Three folders containing searched documents ("PDFs_1_of_3.zip", "PDFs_2_of_3.zip", "PDFs_3_of_3.zip"). *These must be decompressed AND merged into a folder named "PDFs" before use.*
- Two spreadsheets detailing all searched documents ("Document Details.xlsx") and all stopwords skipped during model training ("Stopwords.csv").
- Two Python scripts that execute all functions ("Function_Calls.py") and contain the model functions themselves ("Preprocessing_and_Topic_Modeling_Functions.py").

Table of Contents

List of Acronyms

EPA = Environmental Protection Agency

LDA = Latent Dirichlet Allocation

LMRCC = Lower Mississippi River Conservation Committee

MRB = Mississippi River Basin

NRCS = National Resources Conservation Service

NPS = National Park Service

OCR = Optical Character Recognition

PDF = Portable Document Format

TF-IDF = Term Frequency-Inverse Document Frequency

UMRBA = Upper Mississippi River Basin Authority

UMRCC = Upper Mississippi River Conservation Committee

USACE = United States Army Corps of Engineers

USDA = United States Department of Agriculture

USFWS = United States Fish and Wildlife Service

USGS = United States Geological Survey

Document Search, Selection, and Pre-Processing

Searched Repositories and Selection Criteria

The search for published documents about the Mississippi River Basin (MRB) was limited to those published *after January 1st 1990*. This cutoff was selected for three reasons:

1. Establishing a constraint on the number of documents that inform the model.
2. Documents published prior to this date are more likely to be based on out-of-date research methods, system understanding, and literature.
3. Since the primary goal of this research project is to identify contemporary research priorities of the last 35 years, older studies may not reflect those priorities, or an awareness of the MRB's human-environmental interlinkages.

The phrases *"Mississippi River"* and *"Mississippi River Basin"* were used to search for relevant documents. Documents that used these two phrases in the title, abstract, and/or keywords were added to the list of documents; this criterion ensures that documents are about the MRB, rather than only mentioning the MRB in the main text.

- Other phrases were not searched, because doing so would bias the documents added to the list. For instance, adding "Minnesota" in order to search documents about the MRB performed over Minnesota could add documents to the list that are about Minnesota in a context unrelated to the MRB, due to "Minnesota" appearing in the title/abstract/keywords.

Documents were searched in the following three types of repositories, which are listed below:

- *Federal Repositories:* Federal government websites that keep publicly available reports about the MRB ready for download. Few allow title/abstract/keyword searches to be specified, see Table 1 for further details of the searched repositories.
- *Inter-State Repositories:* Same as above but for basin-scale agencies and non-profit organizations that report on the MRB. See Table 2 for further details.
- *State Repositories:* State government websites that also keep publicly available reports, typically for findings collected over an individual state. The 30 states that overlap with the MRB are selected, with documents typically found through a State Government Website or each state's Department for Environmental Quality, Water Resources, Natural Resources, Fish and Game, Forestry, Agriculture, or equivalent. See Table 3 for further details.
    - *Many results from State Repositories are letters to Federal/State officials, permit requests, data sheets, and news articles for the public that include the MRB in their title/abstract/keywords. These are not reflective of the state of knowledge about the MRB and are thus excluded from the list of documents.*

The initial search for "Mississippi River Basin" in the title/abstract/keywords of documents in these three repositories returned few results, hence the phrase "Mississippi River" was also searched to expand the list of documents.

Table 1: Federal repositories searched for documents about the Mississippi River Basin.

Repository (# Results)	Search Date(s)	Specify Title? (Y/N)	Specify Abstract? (Y/N)	Specify Keywords? (Y/N)	URL	Notes
USGS Publications Warehouse (594)	"Published After: 01/01/1990"	Y	N	N	https://pubs.usgs.gov/	• Publications, reports, and datasets published across journal articles written by USGS scientists. • Seach term field is indifferent to use of quotes or AND statements.
USGS ScienceBase Catalog (304)	N/A	Y	Y	N	https://www.sciencebase.gov/catalog/items/queryForm	• Datasets, reports, ArcGIS products, and other resources from USGS scientists/ affiliates. • Years of publication are not documented consistently must be checked manually. • Category: "Publication" under Advanced Search.
USFWS Publications (14)	"Start Date: Jan 1, 1990"	N	N	N	https://fws.gov/library/categories/publications	• All publications prepared and published by USFWS. • Must search "Mississippi River" in quotes otherwise a much higher number of results returned. • Document Type: "Report" under Refine Your Results.
USDA National Agricultural Library (705)	"Start Date: 01/01/1990", "End Date: 12/31/2023"	Y	N	N	https://search.nal.usda.gov/	• Articles, books, and conferences proceedings produced by USDA and organizations/scientists that received USDA funding. • Search For: "Everything", Material Type: "All Items", Search Filters: "Title contains exact phrase Mississippi River Basin" under Advanced Search.
USDA National Resources Conservation Service (2)	N/A	N	N	N	https://nrcspad.sc.egov.usda.gov/DistributionCenter/default.aspx	• Consists of agency forms and publications prepared by NRCS. • No Advanced Search available.
USACE Digital Library (46)	"Enter Date: after 1990-01-01"	Y	N	N	https://usace.contentdm.oclc.org/	• Reports, letters, maps, and more prepared by the USACE and affiliate organizations. • Collections: “Fish and Wildlife Reports, Histories, IWR Reports, Technical Reports” under Advanced Search.
NPS DataStore (121)	Detailed Dates: “Date after 01/01/1990 AND Date before 12/31/2023”.	Y	N	Y	https://irma.nps.gov/DataStore/Search/Advanced	• Journal articles, reports, factsheets, and dissertations relevant to NPS operations. • Text Fields: “Title contains “Mississippi River” OR Keyword contains “Mississippi River” under Advanced Search. • Documents with the phrase “Mississippi National River and Recreation Area” in their title/keywords are included.
EPA National Service Center for Environmental Publications (13)	Select your archive: tick all from 1991-1994 to 2016-Current.	Y	N	N	https://www.epa.gov/nscep	• Documents, factsheets, and presentations given/prepared by EPA. • Results Precision: "Exact Match" under Field Search.

Table 2: Inter-State repositories searched for documents about the Mississippi River Basin.

Repository (# Results)	Search Date(s)	Specify Title? (Y/N)	Specify Abstract? (Y/N)	Specify Keywords? (Y/N)	URL	Notes
UMRBA Publication Library (48)	N/A	N	N	N	https://umrba.org/library	• Type: "Strategic Plan, Issue Assessment/Reference Document". • Must manually select articles released after 1/1/1990 and before 12/31/2023. • Consists of government documents, strategic plans, and position statements drafted by UMRBA. • The organization's name contains the phrase "Mississippi River", therefore all documents with "UMRBA" or "UMR" in the title are included.
LMRCC Reports (11)	N/A	N	N	N	https://www.lmrcc.org/our-work/reports/	• Consists of a small collection of reports drafted by the LMRCC. • No Advanced Search available. • The lack of search functionality means all reports on the page that use the phrase "Mississippi River" in the title are included.
UMRCC Publications (7)	N/A	N	N	N	https://umrcc.org/our-work/publications/	• Small collection of drafted publications. • No functionality for searching for documents; document with the phrase "Mississippi River" in the title is included.

Table 3: State repositories searched for documents about the Mississippi River Basin.

State	Repository (#Results)	URL	MRB in Title, Abstract, or Keywords?	Mississippi River in Title, Abstract, or Keywords?	Notes
Alabama	Dept. of Environmental Management (0)	https://adem.alabama.gov/default.cnt	N	N	
	Forestry Commission (0)	https://forestry.alabama.gov/Pages/Informational/Search_Results.aspx	N	N	
Arkansas	State Library (0)	https://cdm16039.contentdm.oclc.org/digital/search	N	N	Only State repository with applicable texts.
Colorado	Parks & Wildlife (0)	https://cpw.state.co.us/cpwsearch/pages/results.aspx	N	N	Only State repository with applicable texts.
Georgia	Dept. of Natural Resources (0)	https://gadnr.org/search	N	N	
	Environmental Protection Division (0)	https://epd.georgia.gov/search	N	N	
	State Government Website (0)	https://georgia.gov/search	N	N	
	Water Planning (0)	https://waterplanning.georgia.gov/search	N	N	

Table 3 Cont.

Illinois	Dept. of Natural Resources (13)	https://dnr.illinois.gov/search.html	N	Y	
	Environmental Protection Agency (4)	https://epa.illinois.gov/search.html	N	Y	
	State Government Website (2)	https://www.illinois.gov/search-results.html	Y	Y	
Indiana	State Government Website (0)	https://www.in.gov/core/results.html	N	N	All Department texts accessible through State Government Website
Iowa	Dept. of Agriculture & Land Stewardship (0)	https://iowaagriculture.gov/	N	N	
	Dept. of Natural Resources (3)	https://www.iowadnr.gov/	N	Y	
Kansas	Dept. of Health and Environment (0)	https://www.kdhe.ks.gov/Search	N	N	
	Water Office (0)	https://kwo.ks.gov/required/search-results	N	N	
Kentucky	Dept. of Fish & Wildlife Resources (1)	https://fw.ky.gov/Pages/search.aspx	Y	Y	
	Energy and Environment Cabinet (2)	https://eec.ky.gov/pages/search.aspx	N	Y	
	State Government Website (0)	https://www.kentucky.gov/pages/search.aspx	N	N	
Louisiana	Dept. of Agriculture and Forestry (0)	https://www.ldaf.state.la.us/search_gcse/	N	N	
	Dept. of Environmental Quality (28)	https://edms.deq.louisiana.gov/edmsv2/advanced-search	Y	Y	Documents searched through a central repository.
	State Government Website (5)	https://www.louisiana.gov/	Y	Y	
Maryland	State Government Website (0)	https://www.maryland.gov/pages/search.aspx	N	N	All Department texts accessible through State Government Website
Minnesota	Dept. of Agriculture (0)	https://mn.gov/mda/search/?query=	N	N	
	Dept. of Natural Resources (15)	https://www.dnr.state.mn.us/search?terms=	N	Y	
	State Government Website (8)	https://mn.gov/portal/search/	Y	Y	
Mississippi	Dept. of Environmental Quality (1)	https://www.mdeq.ms.gov/?s=	N	Y	
Missouri	Dept. of Conservation (0)	https://mdc.mo.gov/search	N	N	
	Dept. of Natural Resources (0)	https://dnr.mo.gov/mogov-search/results	N	N	

Table 3 Cont.

Montana	Dept. of Environmental Quality (0)	https://deq.mt.gov/	N	N	
	Dept. of Natural Resources & Conservation (0)	https://dnrc.mt.gov/	N	N	
	Fish, Wildlife & Parks (2)	https://fwp.mt.gov/search	Y	Y	
Nebraska	Idaho Fish & Game (0)	https://dnr.nebraska.gov/search/site	N	N	Only State repository with applicable texts.
New Mexico	Energy, Minerals and Natural Resources Department (0)	https://www.emnrd.nm.gov/sfd/#global-emnrd-search	N	N	
	Game and Fish (0)	https://www.wildlife.state.nm.us/home/nmdgf-site-search	N	N	
New York	Dept. of Environmental Conservation (0)	https://www.dec.ny.gov/searchresult.html#stq=&stp=1	N	N	
	State Government Website (0)	https://search.its.ny.gov/search/search.html	N	N	
North Carolina	Dept. of Agriculture & Consumer Services (0)	https://www.ncagr.gov/search/ag	N	N	
	Dept. of Environmental Quality (0)	https://www.deq.nc.gov/search	N	N	
	Forest Service (0)	https://www.ncforestservice.gov/search.htm	N	N	
	State Government Website (1)	https://www.nc.gov/search/nc	Y	Y	
	Wildlife Resources Commission (0)	https://www.ncwildlife.org/Search-Results	N	N	
North Dakota	Dept. of Agriculture (0)	https://www.ndda.nd.gov/search	N	N	
	Dept. of Environmental Quality (0)	https://deq.nd.gov/search/default.aspx	N	N	
	Water Resources (0)	https://www.dwr.nd.gov/includes/results.php	N	N	
Ohio	Environmental Protection Agency (0)	https://edocpub.epa.ohio.gov/publicportal/edochome.aspx	N	N	Documents searched through a central repository.
Oklahoma	Dept. of Wildlife Conservation (0)	https://www.wildlifedepartment.com/	N	N	
	State Government Website (0)	https://oklahoma.gov/search.html	N	N	
	Water Resources Board (1)	https://oklahoma.gov/owrb/search-results.html	Y	Y	

Table 3 Cont.

Pennsylvania	Dept. of Agriculture (0)	https://www.agriculture.pa.gov/pages/search.aspx	N	N	
	Dept. of Conservation & Natural Resources (0)	https://www.dep.pa.gov/pages/search.aspx	N	N	
	Dept. of Environmental Protection (0)	https://www.dcnr.pa.gov/pages/search.aspx	N	N	
	Game Commission (0)	https://www.pgc.pa.gov/pages/search.aspx	N	N	
	State Government Website (0)	https://www.pa.gov/search/	N	N	
South Dakota	Dept. of Agriculture and Natural Resources (0)	https://danr.sd.gov/default.aspx	N	N	Only State repository with applicable texts.
Tennessee	State Government Website (3)	https://www.tn.gov/search-results.html#tab=department	N	Y	All Department texts accessible through State Government Website
Texas	Commission on Environmental Quality (0)	https://www.tceq.texas.gov/searchpage#gsc.tab=0	N	N	
	Water Development Board (0)	https://www.twdb.texas.gov/search/	N	N	
Virginia	Dept. of Environmental Quality (0)	https://www.deq.virginia.gov/permits/laws-regulations/search	N	N	Only State repository with applicable texts.
West Virginia	Dept. of Environmental Protection (0)	https://dep.wv.gov/Pages/default.aspx	N	N	Only State repository with applicable texts.
Wisconsin	Dept. of Agriculture, Trade and Consumer Protection (0)	https://datcp.wi.gov/pages/SearchResults.aspx	N	N	
	Dept. of Natural Resources (8)	https://dnr.wisconsin.gov/search/google	N	Y	
Wyoming	State Geological Survey (0)	https://www.wsgs.wyo.gov/	N	N	Only State repository with applicable texts.

In addition to Federal/Inter-State/State repositories, a document search was also performed in *Web of Science* to find published scientific articles and books/book chapters about the MRB. The Web of Science search was performed with the following familiar search criteria:

- "Year Published" ranges from 1990 to 2023.
- Title, Abstract, and Author Keywords must contain "Mississippi River Basin". Unlike in the three repositories, the phrase "Mississippi River" was **NOT** searched, since searching "Mississippi River Basin" alone returned a sufficiently large number (800) of documents. Searching "Mississippi River Basin" alone also more closely reflects this project's original intent.

The document search yielded 2,284 unique documents across the Federal/Inter-State/State repositories and Web of Science, of which 2,158 possessed a publicly available full text.

Forming the List of Documents

The file *"Document Details.xlsx"* contains the results of collating this list of 2,284 documents. These documents were collated over the period November 2023 to April 2024. The following information is noted about each document:

- *Title*, *Authors*, *Year,* and *Citation*.
- *Document Type* (Journal Article, Report, Factsheet, Book, Book Chapter, Dissertation, Legislation, Conference Proceedings).
- *Repository Type* (Federal, Inter-State, State, Web of Science). Documents may be locatable in more than repository type.
- *Repository Name* (e.g., USGS Publication Warehouse, LMRCC Reports, Louisiana Department of Environmental Quality, Web of Science). As above, documents may exist in several repositories.
- The *River/Sub-Basin(s)* and *State(s)* over which each document presents information*. A single document may represent multiple Basins and States.
    - Each cell in *River/Sub-Basin(s)* can take one or more of the following values: Arkansas-Red; Basin-Wide; Lower Mississippi; Missouri (Basin); Ohio (Basin); Upper Mississippi.
    - Each cell in *State(s)* can take one or more of the following values: Alabama; All States**; Arkansas; Colorado; Georgia; Illinois; Indiana; Iowa; Kansas; Kentucky; Louisiana***; Maryland; Minnesota; Mississippi; Missouri; Montana; Nebraska; New Mexico; New York; North Carolina; North Dakota; Ohio; Oklahoma; Pennsylvania; South Dakota; Tennessee; Texas; Virginia; West Virginia; Wisconsin; Wyoming.
- Whether *"Mississippi River"* or *"Mississippi River Basin"* appeared in each document's title, keywords, and/or abstract (*MR/MRB in Title?*, *MR/MRB in Keywords?, MR/MRB in Abstract?*).
- Whether the full text is publicly available (*Text Available*).
- Website link to the full text (*URL*).
- The *Preprocessed Text* from all documents in the list.

* If a document's domain of interest intersects with a given spatial scale, it is categorized as such, even if the document does not focus on that domain specifically. For example, a study performed over a single county in West Virginia is categorized as *Ohio (Basin)* under the *River/Sub-Basin(s)* column and as *Indiana* under the *State(s)* column.

** A document is categorized as "All States" under the *States(s)* column if the document's domain covers the entire MRB (*River/Sub-Basin(s)* = "Basin-Wide"), if the document's spatial scale is greater than the MRB itself, or if the spatial scale is unspecified.

*** Documents focused on the Gulf of Mexico are categorized as "Louisiana" under the *State(s)* column and "Lower Mississippi" under the *River/Sub-Basin(s)* column.

Even when a document's domain is smaller than an entire sub-basin, values must still be assigned to both the *States(s)* and *River/Sub-Basin(s)* columns, in order to not exclude documents from model training. For instance, a study performed in Oklahoma will exist within the Arkansas-Red Basin, but a study performed over Minnesota and Wisconsin exists over both the Missouri and Upper Mississippi Basins. Categorizing states by sub-basin is crude since state borders often do not follow sub-basin shapes perfectly. If the states covered in a document were left unspecified, each of the 30 states was thus assigned roughly to each sub-basin in the following manner (states that straddle multiple sub-basins appear more than once):

- *Arkansas-Red Basin*: Arkansas, Colorado, Kansas, Missouri, New Mexico, Oklahoma, Texas.
- *Lower Mississippi Basin*: Arkansas, Kentucky, Louisiana, Mississippi, Missouri, Tennessee.
- *Missouri Basin*: Colorado, Iowa, Kansas, Missouri, Montana, Nebraska, North Dakota, South Dakota, Wyoming.
- *Ohio Basin*: Alabama, Georgia, Illinois, Indiana, Kentucky, Maryland, New York, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia.
- *Upper Mississippi Basin:* Illinois, Iowa, Minnesota, Missouri, Wisconsin.
    - Documents about the *"Middle Mississippi River"*, the river course between St Louis, MO and Cairo, IL, are considered documents with domains over the Upper Mississippi Basin.

The domain of each of the five major sub-basins of the MRB, and the categorization of states by sub-basin, is illustrated below in Figure 1.

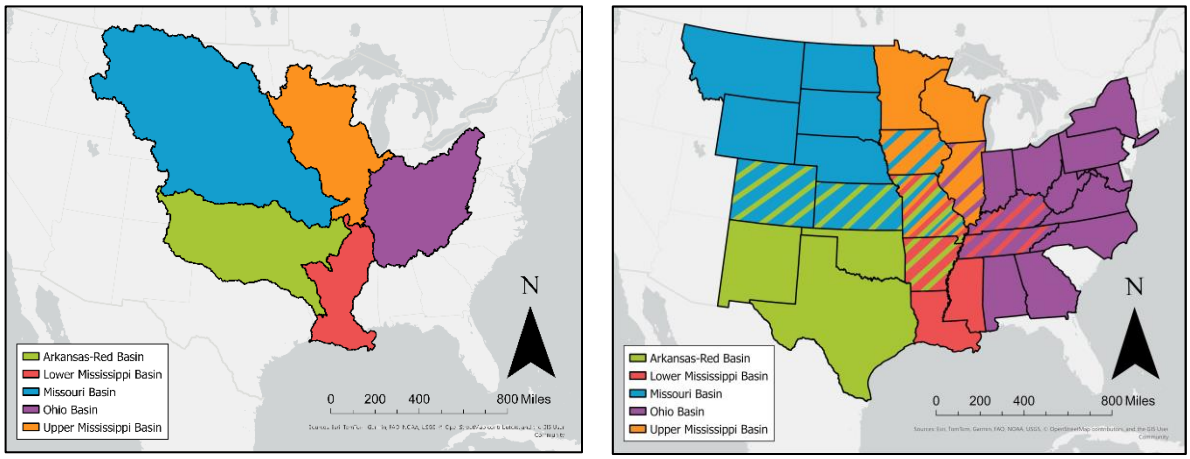

Figure 1: The spatial domains of the five major sub-basins that comprise the Mississippi River Basin (left), and the categorization by sub-basin of the 30 states with which the Mississippi River Basin overlaps (right).
Sub-Basin Shapefile: 2-digit HU (Region) | ArcGIS Hub Home
State Borders Shapefile: TIGER/Line Shapefile, 2021, Nation, U.S., States and Equivalent Entities - Catalog

The purpose of collating such a long list of documents is to represent multiple spatiotemporal scales at which research pertaining to the MRB has been performed. Therefore, a person running this model could select their scale of interest and generate a model output derived from a representatively large number of documents. Listed below are the number of documents that are relevant to each of this study's chosen spatial and temporal scales, of the 2,158 documents that possessed a publicly available full text.

Note that the total number of documents under "Spatial Scales" is greater than 2,158 because the domains of many documents intersect with more than one state or sub-basin.

Spatial Scales – River/Sub-Basin(s):

of *Arkansas-Red* full-text documents: *94*

of *Lower Mississippi* full-text documents: *745*

of *Missouri (Basin)* full-text documents: *201*

of *Ohio (Basin)* full-text documents: *226*

of *Upper Mississippi* full-text documents: *1247*

Spatial Scales – State(s):

of *Alabama* full-text documents: *26*

of *All States* full-text documents: *277*

of *Arkansas* full-text documents: *274*

of *Colorado* full-text documents: *26*

of *Georgia* full-text documents: *15*

of *Illinois* full-text documents: *797*

of *Indiana* full-text documents: *187*

of *Iowa* full-text documents: *582*

of *Kansas* full-text documents: *87*

of *Kentucky* full-text documents: *240*

of *Louisiana* full-text documents: *535*

of *Maryland* full-text documents: *19*

of *Minnesota* full-text documents: *683*

of *Mississippi* full-text documents: *320*

of *Missouri* full-text documents: *665*

of *Montana* full-text documents: *36*

of *Nebraska* full-text documents: *89*

of *New Mexico* full-text documents: *10*

of *New York* full-text documents: *25*

of *North Carolina* full-text documents: *26*

of *North Dakota* full-text documents: *50*

of *Ohio* full-text documents: *85*

of *Oklahoma* full-text documents: *33*

of *Pennsylvania* full-text documents: *39*

of *South Dakota* full-text documents: *141*

of *Tennessee* full-text documents: *215*

of *Texas* full-text documents: *31*

of *Virginia* full-text documents: *22*

of *West Virginia* full-text documents: *38*

of *Wisconsin* full-text documents: *609*

of *Wyoming* full-text documents: *18*

Temporal Scales:

# of full-text documents published in the 1990s (1990-1999): *394*

# of full-text documents published in the 2000s (2000-2009): *549*

# of full-text documents published in the 2010s (2010-2019): *841*

# of full-text documents published in the 2020s (2020-2023): *374*

The *Text Available* column signifies whether a publicly available version of each document existed; 2,158 out of 2,284 documents are thus marked with "Y" in this column (all others are marked with "N"). All documents were downloaded and saved in Portable Document Format (PDF); *these documents must be merged into a folder named "PDFs".* It was, however, necessary to use online tools to "fix" downloaded PDFs under these three circumstances:

- Documents that lacked Optical Character Recognition (OCR), meaning text in downloaded PDFs was not natively machine-readable. This impacted 27 documents, and were indicated with "Y*" under the *Text Available* column.
- Documents that possessed unusual formatting (e.g., each word placed on a separate line upon reading text from a downloaded PDF). This impacted 17 documents, and were indicated with "Y**" under the *Text Available* column.
- Documents that were individual chapters in a larger collection and therefore needed to be extracted (e.g., edited books and conference proceedings). This impacted 13 documents, and were indicated with "Y***" under the *Text Available* column.

Python: Functions used to Pre-Preprocess Documents

Two Python scripts are included alongside this Model Documentation:

1. *"Preprocessing_and_Topic_Modeling_Functions.py"* is used to select PDF files for pre-processing text, extracting main text from each of them, saving pre-processed text to "Document Dataset.xlsx", selecting the texts of interest to the user, and finally enlisting an LDA algorithm to identify and display common topics and words/phrases associated with each. The outcome is a quantitative and visual summary of the topics pertaining to studies of the MRB.
2. *"Function_Calls.py"* is used to run the functions in the above script.

Before attempting to run *"Function_Calls.py"*, please ensure the following:

- Set the PYTHONPATH environment variable as the file path to the folder that contains materials downloaded from GitHub/Open Science Framework. This will allow *"Function_Calls.py"* to access the functions in *"Preprocessing_and_Topic_Modeling_Functions.py"*.
- Make sure to run the lines *import nltk* and *nltk.download('wordnet')* one time in the console window for your current Python environment. Doing this will download NLTK's WordNet database, which you will need for pre-processing the list of documents.
- All modules listed at the top of both scripts must be installed (you will likely need to install the *fitz*, *frontend*, *natsort*, *netgraph*, *PyMuPDF*, and *wordcloud* modules).
- Update the filepath on Line 19 of *"Function_Calls.py"*, so that both scripts can locate the folder named that contains downloaded model materials.
- Any new PDF documents added to the *"PDFs"* folder by the user should have any pages that are categorically not part of the main text be deleted first. This consists of any pages before the main title page, and any pages after the References list (or Acknowledgements or Appendix,

whichever comes first). Any pages that consist only of figures and tables should also be deleted. Online tools can again be used to delete these pages.

- Any new documents that have been added to the "PDFs" folder and fully detailed in "Document Details.xlsx" must also be completed before attempting to pre-process them.

The purpose of pre-processing is to extract the main text from each PDF in the list of documents. The assumption is therefore made that the main text represents the substantive portion of each document and thus conveys its latent topic(s). These pre-processed texts are later fed into the LDA algorithm to identify research topics associated with the selected spatiotemporal scale. **Users may choose to skip the pre-processing by answering "N" to the first user input when running the *"Function_Calls.py"* script.**

Below is a description of the task that each function performs in the pre-processing portion of *"Function_Calls.py"*. Note that these functions frequently rely on user inputs; those that require a user input are underlined.

- *preprocessText*: This function prompts the user to ask whether they wish to redo the main text extraction all PDFs in the list of documents.
    - This is useful should the user alter the pre-processing tasks in *"Preprocessing_and_Topic_Modeling_Functions.py"*, or if new texts have been added to *"PDFs"* and *"Document Details.xlsx".*
    - If the user responds "N", all pre-processing is skipped, and the script moves immediately to LDA algorithm training (see *Python: Functions used to find Latent Topics in the MRB Literature* for details).
    - **IMPORTANT:** *"Document Details.xlsx"* already contains a column called *Preprocessed Text*, which consists of the pre-processed text of all 2,158 documents. Therefore, the user may choose "Y" if they wish not to redo or add any new documents for pre-processing.
- *pdfFileList:* This function creates a list of the names of all PDFs in the list of documents that will be pre-processed for main text extraction.
    - A user input asks whether the user wishes to pre-process only documents that have newly been added to the list, or all of them.
- *pdfToText:* This function is the core component of the PDF to main text pre-processing, which only runs if the user elected to pre-process documents. This function turns each PDF into text and then performs the following tasks on each PDF in the list of documents:
    - Skips headers, footers, and page numbers by drawing a bounding box within which text is extracted from each page.
    - Deletes the Contents page if it exists, along with prior pages. Doing so removes subheadings that would identify sections of the main text. Searches for the first use of the word "Contents" to find the prior pages to be deleted.

- Deletes all text before the Abstract, such as titles, author names and affiliations, and copyright information. All text before the Introduction is deleted instead if no Abstract exists. Searches for the first use of the word "Abstract" in the main text to do so.
- Deletes all text from the References list onward. Doing so removes citations and all subsequent sections, such as Appendices and Acknowledgments. Searches for the last use of the word "References" in the main text to do so.
- Many smaller sections are also deleted separately (including Appendices and Acknowledgments), such as Author Contributions, Data Availability Statements, and Declarations of Competing Interest.

- *delUnwantedLines*: This function deletes remaining lines from the main text extracted by *pdfToText* that are not pertinent to the text's core topic(s). The function iteratively reads each line of the main text, truncates all whitespace, and then deletes entire lines that are the following:
  - Lines only one character in length. These are likely to be page numbers, super/subscript characters, and table cell text.
  - Lines comprised only of numbers, whitespace and punctuation. These are likely the starts of bullet pointed lists, table cells, and comma-separated table cells.
  - Lines containing an "=" sign. These lines are most likely equations and therefore not part of the main text.
  - Lines that contain website/email addresses, phone numbers, and author/university affiliations. These lines are again not part of the main text.
  - Lines that are duplicates of the previous line, most likely being table cells or errors produced in the PDF to text conversion.
  - This function also converts ligature characters into separate Latin characters to ensure the next function does not delete them (e.g., "fl", "fi", "ff", "ffl", "ffi").
- *delInsideLines*: This function deletes unwanted characters without deleting entire lines of text.
  - Numbers, text inside parentheses, hyphens connecting a word across two lines (after joining the word together first), and any non-Latin characters are deleted.
- *tokenizeAndRemove*: This function first deletes all sentences shorter than five characters (left that short by deletion from previous functions) and then tokenizes each main text iteratively to do the following:
  - Delete in-text citations by searching for "et" and "al" as subsequent tokens.
  - Open "Stopwords.csv" and delete all tokens that appear in it. These stopwords consist of singular letters, written numbers one through ninety-nine, equation algebra, common first and last names, common section headings, mathematical units, prepositions, pronouns, conjunctions, common verbs and adverbs (plus conjugations), common adjectives, and common nouns that are non-essential to the main text's arguments.

  - Lemmatization of nouns, verbs, adjectives and adverbs, normalizing all tokens to root forms that simplify their retrieval upon LDA algorithm training.
  - The tokenized text is finally rejoined.

- *appendAndSave*: This function is the final pre-processing step, which will overwrite the existing *Preprocessed Text* column in *"Document Details.xlsx"* with the new pre-processed main text for all redone/newly added documents.
  - The spreadsheet is kept open in the script as a dataframe for use in the LDA algorithm training and evaluation (see *Python: Functions used to find Latent Topics in the MRB Literature*).
- *openDocumentDetails*: This function opens *"Document Details.xlsx"* as a dataframe in preparation to be used for the LDA algorithm training.

There are many unavoidable imperfections in this pre-processing step, and their impact on the accuracy of the topics identified by the trained LDA algorithm should be acknowledged. These imperfections are listed below:

- The words "References", "Contents", and "Appendix" are occasionally used in the main text of documents, therefore cutting off deletion too early (in the case of "References" and "Appendix") or too late (in the case of "Contents"). This reliance on deleting text by identifying section heading names can sometimes cause chunks of main text to be unintentionally lost.
- Tables are often difficult to remove entirely because of inconsistent formatting styles across PDFs prepared by different repositories. While detection of commas, periods and numbers close together can often signify tables, those with cells that contain Latin characters are more difficult if not impossible to detect consistently.
- A table of contents may often summarize multiple chapters with their own "Abstract", "Introduction", "References" (etc.) sections. As such, a table of contents can stretch across multiple pages, with these section headings appearing multiple times, causing a document's main text to occasionally start halfway through a table of contents (e.g., at the first mention of the word "Abstract" that is not on the same page as the word "Contents").
- The bounding box size set to skip headers, footers, and page numbers occasionally skips edges of the main text, since different repositories use headers and footers of different sizes. This inconsistency can cause section deletion to sometimes not work (e.g., the table of contents not being removed if the word "Contents" is above the bounding box).
- Double-column pages are occasionally read as single-column due to a document's OCR. This misreading can create confusing phrases in the extracted main text.
- Of the 2,158 documents with publicly available main texts, three of them lose all of their main text due to the current pre-processing setup (Text IDs 231, 343, and 449).

Applying Latent Dirichlet Allocation to Mississippi River Literature

The Role of LDA in this Project

This project enlists a *topic modeling* approach that facilitates "unsupervised learning that identifies hidden relationships in data"[1], through the application of a Latent Dirichlet Allocation (LDA) algorithm. The objective of an LDA algorithm is to identify hidden ("latent") topics or themes within a corpus of text, without knowing beforehand what those topics are explicitly. LDA algorithms are thus useful when working with large numbers of documents that lack obvious themes or structure, hence they serve to read through text quickly to abstract information in a shorter format. Applications of LDA therefore range from internet search engines to annotating legal documents; anything that requires abstracting topics. In this project's context, LDA is used to identify research topics pertaining to the MRB that are both unique and common to different spatiotemporal research scales (e.g., Upper Mississippi, Kansas, 2000s, etc.).

LDA uses variational Bayes inference[2] (rather than a Markov Chain Monte Carlo method) to probabilistically identify the words most likely to be associated with each topic (word-to-topic), as well as the likelihood of these topics' occurrence within each document (topic-to-document). Each of these two likelihoods is described by an unknown Dirichlet (multivariate-Beta) distribution, which are both constructed by iteratively randomly reassigning topics to words in the list of documents.

Below is a brief description of how the iterations of an LDA algorithm work:

1. Topics are randomly assigned to a seeded sample of words in each document.
    - The number of topics is calibrated beforehand through calculation of coherence and Jaccard similarity metrics.
    - Some iterations may not assign a topic to any words (though unlikely given a large corpus); the use of Dirichlet distributions retains these topics as candidates for future iterations.
2. The probability of words appearing in each topic and the probability of topics appearing in each document are calculated to initiate construction of their two respective Dirichlet distributions.
    - These calculations are based on the presence/absence of words in each document, rather than their total number of appearances. I.e., if the word "flood" appears 20 times in one document, that is still counted as one appearance.
3. One word at a time, topics are reassigned to each word based on the two Dirichlet distributions.
4. Repeat the process as many times as desired.
    - Doing so will cause words to gradually be grouped together under meaningful topics with each subsequent iteration. That is to say, the words most likely to be associated with each topic will "make sense" together.

By iterating an LDA algorithm multiple times over the list of documents for a specific spatiotemporal scale of the MRB, this can model can identify words that commonly occur together within a selected corpus, and therefore abstract its latent topics (*word-to-topic*). Evaluation of the model output against the original PDFs also reveals the documents most likely to be associated with each topic (*topic-to-document*).

Evaluating LDA Performance

Determining whether the output of an LDA algorithm "makes sense" is subjective, because it relies on human inspection of whether the words most likely associated with each topic, and the topics most likely associated with each document, indeed occur frequently in a selected corpus. For instance, assume that the model concludes that the words "sediment", "delta", and "flood" are the three words most likely associated with a topic in 300 documents about MRB research from Louisiana. This result would suggest that sedimentation during flood events in the Mississippi River Delta may be a common research focus, which can be checked against the topic-to-document probabilities and by skimming the original literature (e.g. checking titles, abstracts, and keywords).

Because of how subjective this procedure is, its accuracy can be assured by evaluating an LDA algorithm's performance[3], which this model does in the following ways (the examples below come from running the *"Function_Calls.py"* and *"Preprocessing_and_Topic_Modeling_Functions.py"* scripts over all 2,158 documents):

1. *Word clouds* that summarize the 100 words most likely to be associated with each topic. The size of each word is proportional to its frequency of appearances in each document of a selected corpus, from which the general themes of each topic can be inferred. The example in Figure 2 shows three topics for the MRB's entire corpus, two of which have clearly interpretable topics: one about planning future ecosystem development (Topic 2), and another about river water quality and sediment dynamics (Topic 3).

2. *Word webs* that illustrate the pairwise occurrence of the 25 words most likely associated with each topic. Whereas the word clouds illustrate the frequency at which words appear separately, the word webs provide the added context of words appearing together in pairs. Figure 3 (read clockwise from the right) illustrates pairs of words that occur together more frequently with thicker, more opaque lines, again for the MRB's entire corpus. In Topic 1, words like "calculate", "influence", and "comparison" strongly influence the topic as part of pairs, whereas words like "evidence", "linear", and "capture" strongly influence the topic as individual words. This pairwise examination provides context about which words most commonly occur in the same documents among those that most strongly define each topic.

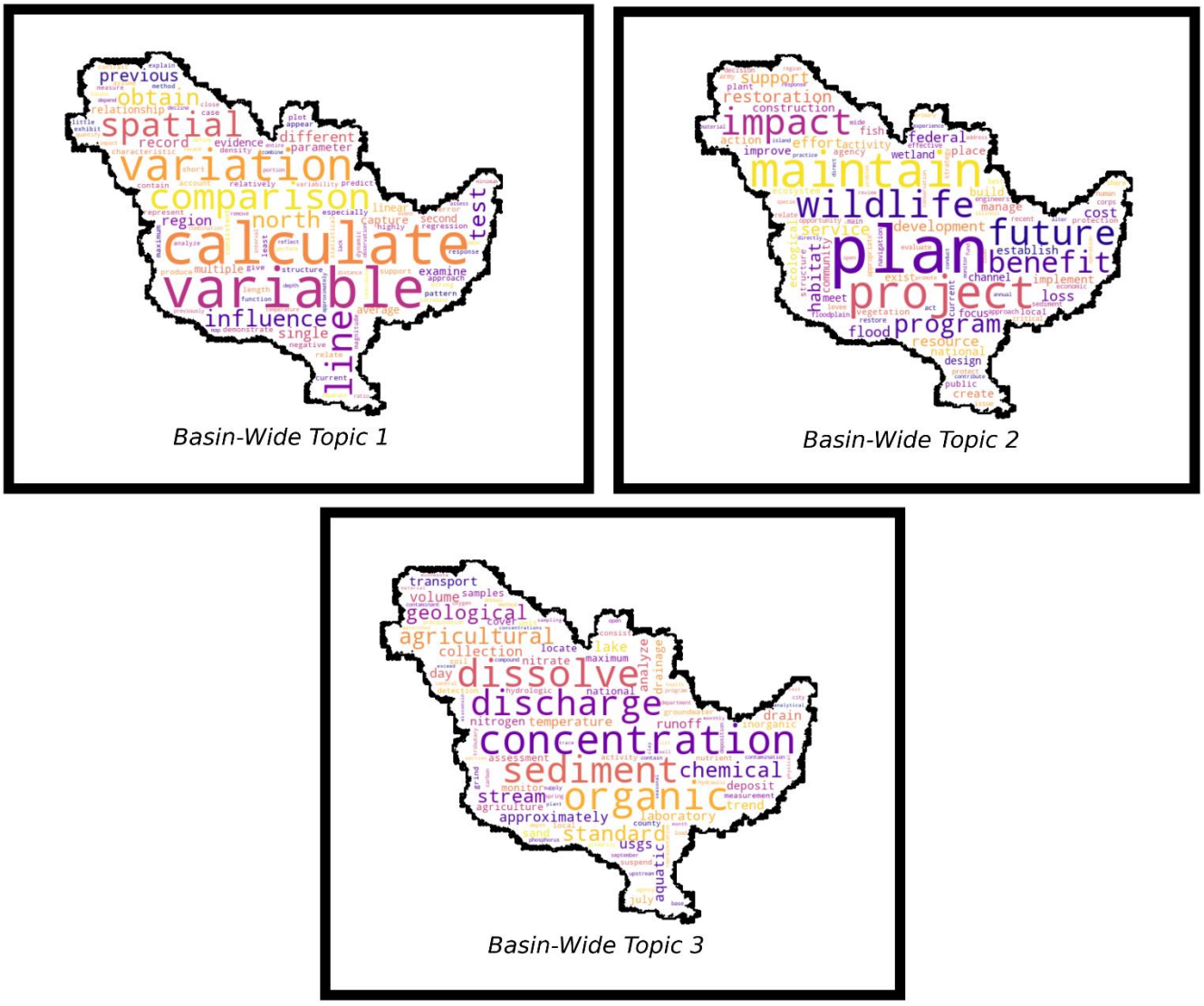

Figure 2: Word clouds produced by training this model's LDA algorithm on all 2,158 documents (scale = "Basin-Wide"). The word clouds are presented in the shape of the selected scale of interest, templates can be found in the "Word Cloud Templates" folder.

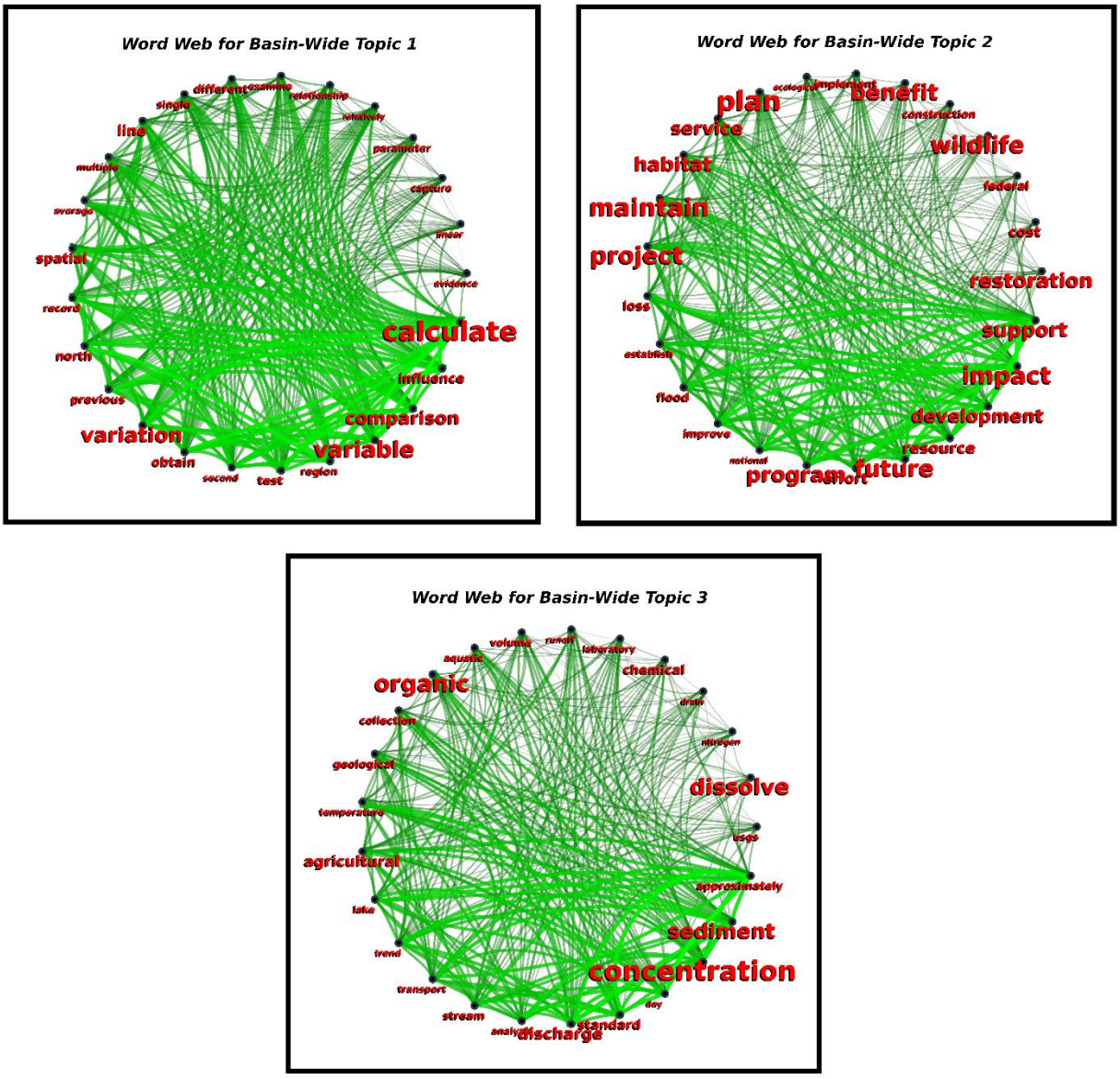

Figure 3: Word webs produced by training this model's LDA algorithm on all 2,158 documents (scale = "Basin-Wide"). Word size is proportional to frequency of occurrence in the documents (as in Figure 2), with thicker, brighter, more opaque lines representing words being those most frequently paired together.

3. *A coherence metric*[4] that summarizes, as a single number, the interpretability of the topics created by the LDA algorithm. Coherence is greatest when the words most likely associated with each topic also occur commonly together as pairs in the same document(s). Greater coherence means that the words in each topic are more like each other, and that the topics are more meaningful[5].
    - Coherence summarizes co-occurrence of words in topics in a standardized manner less subject to human judgment (essentially describing the word webs as a single number). Coherence is a standard means of computing co-occurrence in LDA applications, though it is limited by its inability to convey degree of confidence (no error bars without sensitivity analysis) and its ability to yield "junk topics"[4] (Topic 1 in Figures 2 and 3).
    - The alternative "perplexity" metric is not used by this model, since its value often increases with worsening human interpretability[4], and due to its propensity for yielding large, non-meaningful numbers when dealing with large main texts.
    - While multiple coherence metrics exist, this model uses the recommended UMass score[4], which calculates the log of the probability of pairs of words occurring in a topic, ranging in value from 0 (p=1) to negative infinity (p=0). A score closer to zero means greater coherence.

Between these three evaluation tools, a successful model would produce a high coherence score and topics comprised of high-frequency words that (both individually and in pairs) together suggest MRB research priorities for a given spatiotemporal scale. By maximizing the coherence score (i.e., bringing UMass closer to 0), the meaningfulness of generated topics can be improved, and the model's sensitivity analysis can be performed. There are several parameters in this model that allow for maximizing coherence and performing sensitivity analysis (those that are underlined can be adjusted by the user), each of which are listed below:

- *N-gram Size*: Rather than reading each word in a selected corpus in isolation (i.e., as "unigrams"), the model can additionally read phrases by selecting a maximum n-gram size. Bigrams (n = 2), trigrams (n = 3), and up to decagrams (n = 10) can be added to the model's "bag of words" prior to model training.
    - *Use of larger n-grams means a larger bag of words and thus longer model training.*
- *Removal of the 100 Most Common N-grams*: Although the *"Stopwords.csv"* file does specify common generic n-grams to remove during the pre-processing step, some n-grams that remain are too common across all spatiotemporal scales of MRB research (e.g., river, water, data, basin, site). Retaining these common n-grams causes them to dominate the model output, thus obscuring spatiotemporal distinctness in research priorities, hence the option to remove them.
- *Term Frequency-Inverse Document Frequency (TF-IDF)*: This is a statistical correction[6] applied to all n-grams across an entire selected corpus of documents, which inverse weights n-grams by their frequency of occurrence. N-grams that occur most frequently are thus assigned lower weight prior to the calculation of likelihoods of occurrence of n-grams in each topic.
    - *The application of TF-IDF after also both removing stopwords and deleting the 100 most common n-grams from a corpus can make the computed inverse weights too similar.*

- *The alpha and eta (sometimes called beta) hyperparameters*: These two hyperparameters (definition of hyperparameter here) control the shape of the topic-to-document and word-to-topic Dirichlet distributions, respectively[7]. A greater value for either parameter means that its respective Dirichlet distribution is denser, meaning the trained likelihoods of the most common n-grams (topics) appearing in each topic (document) will typically be closer in value. Smaller alpha (eta) typically means that a smaller number of topics (n-grams) will stand out as being the likeliest to appear within a particular document (topic).
    - *This model is built to calibrate values for the alpha and eta hyperparameters to find those that maximize the coherence score.*
- *Number of Topics*: As with the alpha and eta hyperparameters, this model is calibrated to find the number of topics that maximizes its coherence score. The value of UMass typically approaches zero (i.e., greater coherence) as the number of topics applied to a corpus of text increases.
    - *Having too many topics increases the risk of document overlap, i.e., the same documents being equally likely associated with different topics. The calibration of the number of topics therefore also enlists the Jaccard similarity index to minimize document overlap while maximizing coherence[8]. Since Jaccard similarity and UMass have different scales, both must be normalized to facilitate comparison.*
- *Seed of Randomly Sampled Words:* The final parameter that is calibrated by the model to maximize its coherence score. The first step of the LDA algorithm training seeds a number of n-grams to be randomly assigned topics across all documents in the selected corpus. This parameter controls the seeded number of n-grams in each iteration.

The results of calibrating these final four (hyper)parameters (alpha, eta, number of topics, seed size) are output by the model prior to its evaluation. An example output again using all 2,158 documents to train the model is shown in Figure 4. This figure illustrates the range of values that are trialed by the LDA algorithm, with each one varied separately while the others are held constant. The black dashed line in each chart conveys the calibrated value that maximizes coherence (and also minimizes Jaccard similarity in the case of number of topics), each of which are used by the LDA algorithm for the run that produces its definitive results for the corpus. Producing and comparing different versions of Figures 2, 3, and 4 by modifying all of these (hyper)parameters allows sensitivity analysis to be performed, i.e., how the topics yielded by this model change in response to changes in (hyper)parameters.

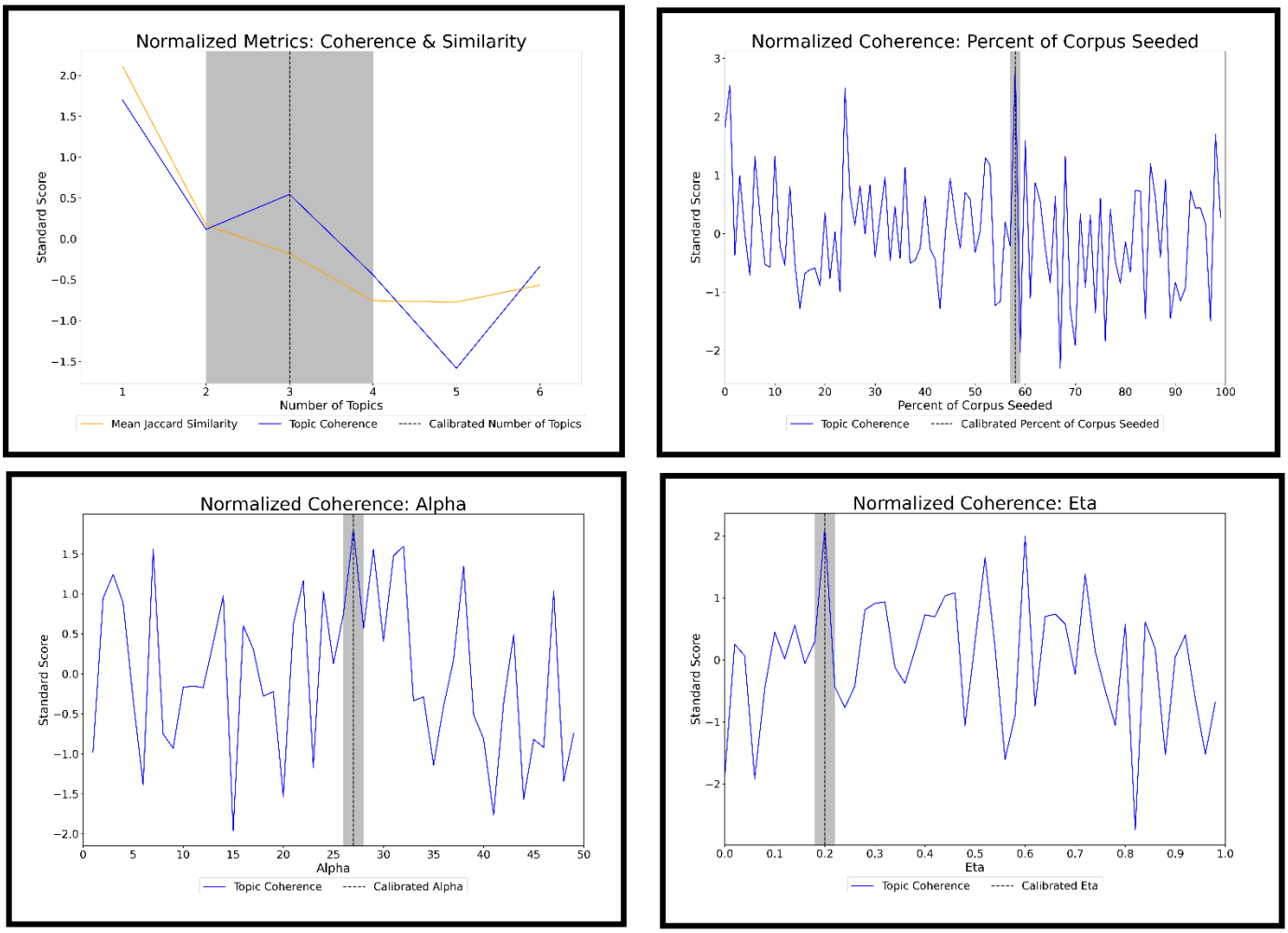

Figure 4: Example results of calibrating (hyper)parameters with this model's LDA algorithm. From top-left to bottom-right are the results for the number of topics, seed size (expressed as a percentage), alpha, and eta (hyper)parameters. Black dashed lines indicate the calibrated values used in the algorithm's definitive model run. Produced by training this model's LDA algorithm on all 2,158 documents (scale = "Basin-Wide").

Python: Functions used to find Latent Topics in the MRB Literature

Following on from *Python: Functions used to Pre-Preprocess Documents*, the second half of the *"Preprocessing_and_Topic_Modeling_Functions.py"* and *"Function_Calls.py"* scripts calibrate, train, and evaluate the LDA algorithm using the pre-processed main text. Below is a description of all of the functions in this half of the script, how they each work and an explanation of the outputs that each one produces. As before, functions that require a user input are underlined.

- *openDocumentDetails*: This function opens *"Document Details.xlsx"* as a dataframe if the user chose **not** to redo the pre-processing of all PDFs in the list of documents (see *Python: Functions used to Pre-Preprocess Documents*).
    - This is also the final function used in the pre-processing half of the scripts.
- *textSelection*: This function specifies the spatiotemporal scale from which main texts will be used from the *Preprocessed Text* column of the dataframe to train the LDA algorithm.
    - The user is prompted to decide the scope of texts they wish to use: all 2,158 texts ("*All*"), texts from a specific basin ("*Sub-Basin*"), texts from one of the 30 states ("*State*"), or texts from a specific decade ("*Decade*").
    - If the user does not select "*All*", then a second prompt appears asking for a specific sub-basin, state, or decade from which to extract the pre-processed text. For instance, if the user selects "*Decade*" and then "*2000s*", the dataframe is sliced down to rows corresponding to documents published between the years 2000 and 2009.
- *createCorpus*: This function takes all of the selected text from the dataframe's *Preprocessed Text* column and creates a corpus for training the LDA algorithm.
    - The user is first prompted to select the **maximum n-gram size of interest**, ranging from 1 to 10. For instance, if 2 is selected, the text will be read as unigrams and bigrams, meaning the created corpus will consist of every individual word, plus all the same words as pairs.
    - The user is additionally prompted to decide whether to **remove the 100 most common n-grams** found across all 2,158 documents. If the user responds with "Y", all instances of the following n-grams are removed from the created corpus (in order of frequency of occurrence):

      river, mississippi, mississippi river, water, area, high, low, large, data, time, increase, analysis, result, system, range, year, indicate, great, occur, change, long, level, present, similar, determine, number, value, small, location, report, condition, compare, effect, period, first, represent, upper, different, measure, average, estimate, site, important, available, significant, term, flow, information, limit, reduce, process, state, difference, associate, potential, source, point, remain, surface, describe, observe, identify, sample, collect, factor, describe, develop, rate, vary, mean, natural, basin, affect, major, model, control, likely, scale, example, decrease, cause, distribution, reach, size, relatively, early, environmental, type, management, select,

new, generally, field, survey, relative, require, specific, research, land, quality, individual.

- To illustrate the 100 most common n-grams that remain in the created corpus, a word cloud is made and then saved to the respective folder directory in "Model Training Results".

- *trainLDAAlgorithm*: This function is the core of the entire model and is used to calibrate the LDA algorithm's parameters and then train itself on the created corpus. This model uses the LDA algorithm functions in Python's gensim module[9] (the LDA algorithm in Python's sklearn module was not used due to lacking a native function for calculating coherence[10]) to do the following:
  - The user is prompted to answer whether they wish to use a **TF-IDF correction** to inverse weight n-grams before using them to create a "bag of words". If the user responds with "N", the bag of words is created without this correction.
  - The four (hyper)parameters, **number of topics**, **seed**, **alpha**, and **eta** are then calibrated by passing the bag of words over the LDA algorithm 50 times, varying one (hyper)parameter while keeping the other three constant.
    - **Number of topics** is calibrated first, using the default values given in gensim's LDA function for the other three (hyper)parameters. Since one of the limitations of using coherence scores to determine meaningfulness of topics is the creation of "junk topics"[4], this model limits the maximum calibrated **number of topics** to six. Otherwise, the calibrated **number of topics** is that which maximizes coherence (based on UMass score) and minimizes similarity (based on Jaccard similarity index), both of which are normalized for comparison.
    - **Seed size** (named *random_state* in gensim's LDA algorithm function) is calibrated using this same **number of topics** (again holding **alpha** and **eta** as default while not being calibrated). The tested values of **seed size** range from 0 to the number of n-grams in the created corpus (in percentiles from 0 to 100). The **seed size** value that maximizes coherence is used moving forward.
    - According to this function's documentation[9], and several published works[11,12,13], **alpha** is often parameterized as 50/(**number of topics**), and **eta** as 0.1. The user has the option of using these default values from the literature, or calibrating values for them that maximize coherence. Calibration values tested range from 1 to 50 (in increments of 1) for **alpha**, and 0 to 1 (in increments of 0.02) for **eta**.
  - The calibration results are plotted (see Figure 4 for an example), then the four calibrated (hyper)parameters and the bag of words are again passed 50 times over the LDA algorithm to produce the final model outputs.

- *evaluateTrainedModel:* This function produces charts and tables that interpret the LDA algorithm's effectiveness at assigning n-grams to likely topics and topics to likely documents. All of these outputs are again added to the respective folder directory in "Model Training Results".

- A *word cloud* is created for the 100 n-grams most likely to be associated with a topic, with one word cloud produced per topic (see Figure 2 for an example). **This summarizes the word-to-topic Dirichlet distribution.**
- *Document-topic densities* are calculated for each document that was selected to train the LDA algorithm. These densities convey the likelihood of each topic best describing the contents of each document; the topic with the greatest likelihood is assumed to be that which best describes the document. **This summarizes the topic-to-document Dirichlet distribution** and is saved as a spreadsheet named *"Document-Topic Densities.csv"*.
- A *word web* is created for the 25 n-grams most likely to be associated with a topic, with one word web produced per topic (see Figure 3 for an example). **These provide visual context to the computed coherence score, since both describe co-occurrence of n-grams within a topic.** The word web outputs for each topic are also produced in spreadsheet form, named *"Spreadsheet for Word Web for Topic *number here*.csv".*

- *writeTextFile:* This function writes a text file that summarizes all of the user input decisions made (e.g., whether a TF-IDF algorithm was used, the selected n-gram size, the decision to calibrate alpha and eta or not), along with a few quantitative results:
  - The number of documents in which the 100 most common n-grams appear.
  - The calibrated (hyper)parameter values.
  - The UMass coherence score of the trained model.
  - This file is written to the *"Model Training Results"* folder with the name *"End-User Decisions and Other Outputs.txt"*.
- *moveToSubFolder:* This function packages all of the created charts, spreadsheets, and text file into a unique sub-folder. The sub-folder's name reflects the end-user decisions made to produce it, allowing for model runs with different end-user setups to be saved and stored.

References

1. High Demand Skills (2022). Topic Modeling with LDA Explained: Applications and How IT Works. Accessed January 8th 2025. https://highdemandskills.com/topic-modeling-intuitive/.
2. Hoffman MD, Blei DM, Bach F. (2003). Online Learning for Latent Dirichlet Allocation. Proceedings of the 24th International Conference on Neural Information Processing Systems – Volume 1. Vancouver, BC, Canada. 856-864. https://proceedings.neurips.cc/paper_files/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf.
3. High Demand Skills (2022). Topic Model Evaluation. Accessed January 8th 2025. https://highdemandskills.com/topic-model-evaluation/.

4. Zvornicanin E, Martin E. (2024). When Coherence Score is Good or Bad in Topic Modeling? Accessed January 8th 2025. https://www.baeldung.com/cs/topic-modeling-coherence-score.
5. Tresnasari NA, Adji TB, Permanasari AE. (2020). Social-Child-Case Document Clustering based on Topic Modeling using Latent Dirichlet Allocation. Indonesian Journal of Computing and Cybernetics Systems, 14(2), 179-188. https://doi.org/10.22146/ijccs.54507.
6. GeeksForGeeks (2023). Understanding TF-IDF (Term Frequency-Inverse Document Frequency). Accessed January 8th 2025. https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency.
7. Welbers K. (2025). Dirichlet distributions, the alpha hyperparameter, and LDA. Accessed January 8th 2025. i.amcat.nl/lda/understanding_alpha.html.
8. StackOverflow (2015). What is the best way to obtain the optimal number of topics for a LDA-Model using Gensim? Accessed January 8th 2025. https://stackoverflow.com/questions/32313062/what-is-the-best-way-to-obtain-the-optimal-number-of-topics-for-a-lda-model-usin.
9. Gensim (2023). Latent Dirichlet Allocation. Accessed January 8th 2025. https://radimrehurek.com/gensim/models/ldamodel.html.
10. StackOverflow (2020). How do I calculate the coherence score of an sklearn LDA model? Accessed January 8th 2025. https://stackoverflow.com/questions/60613532/how-do-i-calculate-the-coherence-score-of-an-sklearn-lda-model.
11. Griffiths, TL, Steyvers M. (2004). Finding scientific Topics. PNAS Colloquium, 101, 5228-5235. https://doi.org/10.1073/pnas.0307752101.
12. Steyvers M, Griffiths TL. (2007) Probabilistic Topic Models. In T Landauer, D McNamara, S Dennis, W Kintsch (Eds.) Latent Semantic Analysis: A Road to Meaning. 22pp. Psychology Press.
13. Grün B, Hornik K. (2011). topicmodels: An R Package for Fitting Topic Models. Journal of Statistical Software, 40(13), 1-30. https://doi.org/10.18637/jss.v040.i13.