DSSS Assignment 1

Yukun Jiao

May 15, 2025

Introduction and Ethics

The aim of this project is to obtain metadata of articles published across 8 volumes in *Journal of Computational Social Science*[1] and to fit an LDA topic model with 20 topics to the corpus of abstracts from these articles, in order to estimate the topic of each volume. The data can be obtained either by crawling and scraping all article list pages and the corresponding article URLs to extract metadata from each article's HTML content, or by using the API to directly retrieve metadata for specific article URLs. The LDA modeling and analysis process follows the illustrative example "Abstracts of JSS papers" provided in the topicmodels package.[2]

Considering ethics, I first checked whether Springer provides an API for accessing article metadata, since using an API is more ethical and legitimate than scraping HTML pages. Fortunately, Springer provides an API for research purposes (I included the API key in my R script).[3] By supplying the article URLs, I was able to access the corresponding article metadata through the API.

Subsequently, I reviewed Springer's Terms of Use.[4] According to the publisher Springer's Terms of Use (Section 1.3), systematically downloading content or continuously and automatically indexing metadata is not permitted. This means it would be inappropriate to use a crawler to collect large amounts of article metadata or to download many PDFs.

I then examined SpringerLink's robots.txt.[5] Fortunately, it allows crawlers to access certain article-related resources (e.g., /article$, /article/, /articles/$), which means I am allowed to use crawlers to retrieve article URLs.

Therefore, in this study, all metadata used for analysis was accessed via the API by providing the corresponding article URLs. To ensure compliance with the API's rate limits, all requests followed the restrictions of no more than 100 requests per minute and 500 requests per day.

[1] link.springer.com/journal/42001
[2] 10.18637/jss.v040.i13
[3] dev.springernature.com
[4] link.springer.com/termsandconditions
[5] link.springer.com/robots.txt

Additionally, for the purpose of demonstrating what I learned in the DSSS course, I used web scraping methods (without using the API) to scrape metadata for two articles and download their PDFs, with a 2-second delay between each request. This was done with a clear user-agent string (DSSS student project/1.0 (MacOS; only for course assignment; contact: yukji739@student.liu.se)) to identify the purpose of the requests.

Data Crawling and Scraping

At the beginning, a headless browser was used to access the homepage of the *Journal of Computational Social Science* as well as the dynamically loaded article list page and the login page. By controlling the headless browser to perform clicking and input actions, I successfully interacted with multiple webpage elements, achieving login and entry into the article list page. By extracting the pagination information, I determined that the page has a total of 8 pages, containing 377 articles (on May 10, 2025).

Subsequently, I crawled all article URLs from the article list pages and saved them for later use with the API. For demonstration purposes, I selected two article URLs, crawled the corresponding pages, extracted article metadata using XPath, and downloaded the PDFs. I used a regular expression to remove illegal characters from the article titles and used the cleaned titles as filenames when saving the PDFs. The metadata of these two articles was saved in a data frame and exported as a CSV file.

Finally, I used the collected article URLs to access article metadata via the API. The retrieved metadata was in JSON format. I saved these JSON-format metadata as "api_responses.rds" to avoid repeated API access. I then applied regular expressions to extract the metadata of interest (abstracts, year, title, authors, and volumes), bound the results into a data frame, and exported it as a CSV file for further analysis.

Analysis

A mini analysis was conducted on the collected metadata to identify key topics and words across the volumes. First, I converted the documents in the final dataset into a corpus, and then exported the corpus as a document-term matrix. The mean term frequency-inverse document frequency (tf-idf) over the documents containing each term was used to select the vocabulary. I only included terms that had a tf-idf value of at least 0.1.

Then, I fit an LDA model with 20 topics using VEM with estimated α based on the document-term matrix with a reduced vocabulary.

For each volume (from 1 to 8), the most possible topic was determined, and the 10 most frequent terms associated with the topic were extracted. These results were shown in Table 1 and Table 2, presenting the top topic terms for each volume.

Table 1: The Ten Most Frequent Terms for Each Volume (Part 1)

Volume 1	Volume 2	Volume 3	Volume 4
flow	team	bot	urban
graph	monetari	incom	crime
firm	reward	persuas	dropout
sale	clickbait	debat	climat
incom	judgment	disinform	citi
protest	reput	infodem	crimin
inequ	workflow	parliamentari	blockchain
app	spoiler	journalist	miss
inflat	cgm	affin	satisfact
bdi	flock	african	secur

Table 2: The Ten Most Frequent Terms for Each Volume (Part 2)

Volume 5	Volume 6	Volume 7	Volume 8
diffus	hate	emot	citi
forum	speech	vaccin	llms
stanc	fake	disast	controversi
frame	spcr	parti	contagion
toxic	abm	immigr	reddit
bridg	antisemit	antivaccin	gpt
refuge	xai	coupl	tenant
extremist	cancer	fear	cascad
fertil	disast	romant	crowd
stereotyp	slaveri	deepconnect	evacu