Assignment 3: Text and Network Visualization with ArXiv Papers

Thomas Brambor

Text and Network Visualization with ArXiv Papers

Data

ArXiv is a free distribution service and open-access archive for scholarly articles, primarily in physics, mathematics, computer science, and related disciplines. For this assignment, we'll be working with a dataset of recent papers from the "Physics and Society" (physics.soc-ph) category, which covers research on social phenomena, collective behavior, opinion dynamics, social networks, and other topics at the intersection of physics and social science.

The dataset contains information about approximately 10,000 recent papers published between 2018 and the present. This subset has been extracted from the complete ArXiv dataset.

`arxiv_subset.csv`

variable	class	description
id	character	ArXiv paper ID
title	character	Paper title
authors	character	Author names (comma-separated list)
categories	character	ArXiv categories (papers can belong to multiple categories)
update_date	character	Last update date
abstract	character	Full text abstract of the paper

Additional information is available in the full JSON file (`arxiv_subset.json`) including version history, DOI, and other metadata. The JSON file also contains the parsed author names, making it easier to work with author data.

Tasks

In this assignment, you will create a series of visualizations to explore the ArXiv papers dataset through text and network analysis techniques. Your notebook should be well-documented with explanations of your approach and interpretation of results. **Choose two of the three main tasks below to complete** (Text Analysis, Author Network, or Topic Analysis).

1. Text Analysis and Visualization

In this section, you'll explore the textual content of research papers through visualization techniques. Use the full dataset of 10,000 papers for this analysis.

a) **Term Frequency Analysis**

- Prepare the paper abstracts by removing stopwords, punctuation, and numbers
  - *Hint: In R, the `tidytext` package provides functions like `unnest_tokens()` to tokenize text; in Python, use `nltk` or `spaCy` for tokenization and stopword removal.*

- Create a bar chart showing the top 20 most frequent terms across all abstracts
  - *Hint: After tokenizing, count term frequencies and sort to find the most common terms.*
- Generate a comparative visualization (e.g., small multiples or grouped bars) showing how the top 10 terms differ between the years 2019 and 2023
  - *Hint: Use `update_date` as the source of the paper year.*

b) **Word Cloud and Term Co-occurrence** (Choose one)

- Option 1: Generate a word cloud of the 100 most frequent meaningful terms in the abstracts, with proper color encoding and layout.
  - *Hint: Use the `wordcloud` package in R or `wordcloud/matplotlib` in Python to create visualizations with control over colors and sizes.*
- Option 2: Create a network visualization showing the 30 most frequent terms and how they co-occur within abstracts.
  - *Hint: Find terms that appear together in the same document, then visualize as a network where edge weight represents co-occurrence frequency.*

2. Author Collaboration Network

In this section, you'll analyze and visualize the network of research collaborations:

a) **Network Construction**

- **Filter the dataset** to include only papers that are categorized as both "physics.soc-ph" (Physics and Society) AND "cs.SI" (Social and Information Networks)
- Extract author information from these papers and create a network where authors who co-authored at least one paper are connected
  - *Hint: For each paper, create pairs of co-authors. These pairs will form the edges of your network.*
- **Further limit your analysis to authors who have published at least 4 papers** in this subset to keep the visualization manageable
  - *Hint: Count papers per author, then filter to keep only prolific authors.*

b) **Network Visualization**

- Create a visualization of the author collaboration network using an appropriate layout algorithm
- Size nodes based on the number of papers published by each author
  - *Hint: Map the paper count to node size in your visualization.*
- Color nodes based on a relevant metric (e.g., betweenness centrality or degree)
  - *Hint: Use centrality metrics available in network analysis packages like `igraph`.*
- Label the 10-15 most prolific authors

3. Topic Analysis Using LLMs

In this section, you'll use the Large Language Model approach we discussed in class to analyze topics in the papers:

a) **Topic Identification**

- Select a random sample of at least 200 paper abstracts from the dataset (no need to go overboard as larger size will slow donw the task and run into rate limits of free API tiers)
- Use an LLM (e.g., Google's Gemma, OpenAI, Anthropic Claude, or a local model) to analyze each abstract and assign topic labels
  - *Hint: Create a clear prompt that asks the LLM to identify specific research topics in the abstract. For example: "What are the 2-3 main research topics in this scientific abstract?"*
  - *Hint: For consistency, use the same prompt for all abstracts and request a specific format for the response (e.g., comma-separated topics or a `json` structure).*
- Create a visualization showing the distribution of these assigned topics

b) **Topic Trends**

- Group papers by year and visualize how the prevalence of different topics has changed over time
- Create a visualization showing topic evolution over time
- Identify and discuss any noticeable trends and explain how your graph design choice emphasize understanding the main pattern(s)

Implementation Tips

1. **Start simple**: Focus on creating clear, informative visualizations rather than complex ones. A well-executed simple visualization is better than a poorly executed complex one.
2. **Text processing**: For tokenization and text cleaning, use packages like `tidytext` in R or `nltk`/`spaCy` in Python. Focus on basic techniques like stopword removal and normalization.
3. **Network analysis**: Use `igraph`/`ggraph` in R or `networkx` in Python for network construction and analysis. Both provide similar functionality for calculating metrics and creating visualizations.
4. **LLM API usage**: For the topic analysis, you can use Google's Gemma (as demonstrated in class), OpenAI's API, Anthropic's Claude, or any other accessible LLM.
5. **Documentation**: Briefly explain your visualization choices and what insights they reveal about the data.

Submission

Your submission should include:

1. A Jupyter notebook, Quarto, or R Markdown document with your code, visualizations, and analysis
2. A rendered HTML version of your document

Please follow the instructions to submit your homework. The homework is due on Monday, April 16.

Please stay honest!

If you do come across something online that provides part of the analysis / code etc., please no wholesale copying of other ideas. We are trying to evaluate your abilities to visualize data not the ability to do internet searches. Also, this is an individually assigned exercise – please keep your solution to yourself.