Analyzing Wikipedia collaboration networks
BA-/MA-Projects

Jürgen Lerner

University of Konstanz

Summer 2017
(last updated: April 26, 2017)
Wikipedia: encyclopedic articles written by volunteers.

How do Wikipedians organize themselves?

What explains their productivity or output quality?
Why studying Wikipedia?

Wikipedia is an organization in which actors are jointly creating products that are intended to have appeal to others.

*Facilitates the analysis of team productivity.*

Studying Wikipedia has some outstanding positive aspects:

- large-scale, fine-grained, and complete data availability;
- clearly defined intended outcome & indicators for success;
- relevance of the created products (people are using them);
- novel form of organization (self-governance, volunteers).

⇒ Studying Wikipedia is convenient and relevant.
What are we doing in this project?
Outline of this project.

Write software to
- collect parts of the Wikipedia database;
- combine, filter, preprocess this data;
- performing a preliminary analysis (descriptive, visual).

**Deliverables** of each participant:
- documented program code;
- results of preliminary analysis and/or descriptive statistics of collected data;
- short presentation at the end of the lecture period.
IMPORTANT: To-do until next week.

Topic assignment:

- Participants send me per email until Friday, April 28 a preference list of the topics:
  1. most preferred topic
  2. . . .
  8. least preferred topic

- I select randomly from those most interested in a topic and post these assignments on the webpage before Wednesday next week.

If you want to participate but have no preferences, send an email without the list.
Tentative schedule.

Until Friday, April 28: send topic preference-list to me.

May 3: additional clarification about assigned topics / suggested way to proceed.

Until May 24: tasks and work plan should be precisely clarified.

Presentations (15–20 Min) of own project July 19 and 26.
Topics.

(Your are invited to provide your own ideas.)
Featured article candidate (FAC) assessment.

*Feature article* is the highest quality label for Wikipedia articles.

- Users can **nominate** *featured article candidates*.
- Users can discuss, **support**, **oppose**, these nominations.
- Articles are promoted if there is (positive) **consensus**.

Featured article candidate nomination, discussion, and voting are fully transparent.

- Featured articles can be **demoted** if they no longer meet the criteria.

(1) Featured article candidate (FAC) assessment.

**Task:** collect data on nominations, votes, and outcome: who contributes when and how in the evaluation of which article.

E. g., \((time_i, user_i, article_i, suggest | support | oppose, \ldots)\)

**Exemplary empirical questions**

- who participates in FAC evaluation (established users vs. novice, authors of the article vs. bystanders, \ldots)?
- who nominates featured article candidates?
- are supporters systematically different from opponents?
- are there specialized roles in the FAC process (checking references, assessing neutrality, correctness, \ldots)?
- are there influential users? do users follow others?
- descriptive statistics
Requests for adminship (RFA) assessment.

Wikipedia administrators have special rights such as user blocking, page protection, page deletion.

- Users can **(self-)nominate** candidates.
- Users can discuss, **support**, **oppose**, these nominations.
- Users are promoted if there is (positive) **consensus**.

Requests for adminship nomination, discussion, and voting are fully transparent.

(2) Requests for adminship (RFA) assessment.

**Task:** collect data on nominations, outcome and votes: who contributes when and how in the evaluation of which user.

\[(time_i, voter_i, candidate_i, suggest | support | oppose, \ldots)\]

**Exemplary empirical questions**

- who participates in RFA evaluation (established users vs. novice, users often collaborating with the candidate, \ldots)?
- are candidates nominated by themselves or by (which) others?
- are supporters systematically different from opponents?
- are there specialized roles / recurrent types of questions to the candidate in the RFA process?
- are there influential users? do users follow others?
- descriptive statistics
User interaction through co-editing articles.

The text of a Wikipedia article can be modified by many users at many point in time (versioned documents).

When modifying the text of an article, users can

1. provide new content;
2. undo contributions of (which) others;
3. restore previously deleted (by whom) contributions (whose).

Gives information about the article production process.
(3 & 4) User interaction through co-editing articles.
(3 & 4) User interaction through co-editing articles.

**Task:** write software to capture the fine-grained, typed, time-stamped user interaction from text modifications.

- **Topic 3:** by appropriately computing **text differences**.
- **Topic 4:** using appropriate **vector representations** (e.g., tf-idf) of documents.

**Exemplary empirical questions**
- who has written which part / how much of the text
- whose text remains on the page
- detection of roles of contributors (content providers, spell-checkers, vandal fighters, . . .)
- life-cycle of pages: content creation, stabilization;
- explain article quality by co-editing network;
- descriptive statistics
(5) User interaction through discussion pages.

Users can discuss the article creation in associated talk pages.
(5) User interaction through discussion pages.

**Task:** write software to encode user interaction from discussion on talk pages.
- who starts discussion threads; who replies to whom?
- potentially: identify types of contributions (e.g., references to Wikipedia policies, suggesting changes to the article, pointing out miss-behavior, . . . )
- potentially: sentiment analysis (positive/negative; sentiment targeted to what or whom)

**Goals / research questions:**
- develop standardized, automatically extractable way to encode discussion
- identify relations between contributors of articles and associated talk pages
- identify relations between discussion and article quality
(6) Edit comments.

Users can (and should) provide comments when they upload new revisions of an article.

**Task:** write software to encode edit comments.

- identify types of comments (including references to Wikipedia policies, references to other users, mentioning types of edits, ...)
- potentially: sentiment analysis (positive/negative comments; sentiment targeted to what or whom)

**Goals / research questions:**

- develop standardized, automatically extractable way to encode edit comments
- develop typology of edit comments
- relation between comments and article quality / edit stability
(7 & 8) Attention and activity.

Wikipedia articles receive varying amounts of attention.

Wikipedia contributors show varying levels of activity.
Attention dynamics of articles.

Timepoints of all edits to all articles are known exactly.

**Task:** perform a basic analysis of the dynamics of attention received by (some) articles. Either by analyzing the duration between edit events or the timeseries of the number of events in given intervals.

**Goals / research questions:**
- identifying general trends and/or periodicity;
- are there types of pages with similar attention dynamics (steady editing, periodic editing, bursty editing);
- are there inter-dependencies between the attention dynamics of different pages;
- what makes some pages more popular than others.
Activity dynamics of contributing users.

Timepoints of all edits of all contributors are known exactly.

**Task:** perform a basic analysis of the dynamics of activity of (some) contributors. Either by analyzing the duration between edit events or the timeseries of the number of events in given intervals.

**Goals / research questions:**
- can you identify general trends and/or periodicity;
- are there types of users with similar activity dynamics (steady editing, periodic editing, bursty editing);
- are there inter-dependencies between the activity dynamics of different contributors;
- what explains the level of activity of contributors.
Wikipedia: data.
Core structure: network connecting users with pages.

A lot of additional information attached to users, pages, links, ...
General Wikipedia data sources.

Information about Wikipedia database download

Database schema (table names and fields)
https://www.mediawiki.org/wiki/Manual:Database_layout

Finding the actual files https://dumps.wikimedia.org
- click on Database backup dumps
- select the wiki of interest (e.g., enwiki)
- if the dump is in progress or failed: Last dumped on . . .
A social network consists of a few dozen to at most 150 people organized in social relationships and [[weak friendship]]s ...

A social network consists of any group of people connected through various social ...

...

The XML files encode revisions of pages (some the **current state**, some the **whole history**).


- `stub-meta-history.xml.gz`  
  whole page history; except the pages’ text
- `pages-meta-current.xml.bz2`  
  only the last revision, including text
- `pages-meta-historyX.xml-pYpZ{.7z|.bz2}`  
  whole page history, including text
Downloading the history of single (or few) pages.

Up to 1000 revisions can be downloaded from https://en.wikipedia.org/wiki/Special:Export

WikiEvent: download the complete history of few pages: http://algo.uni-konstanz.de/software/wikievent/

Wikimedia API can be used for download and editing: https://www.mediawiki.org/wiki/API:Main_page
Wikipedia database dumps: SQL files.

The SQL files (ending with sql.gz; at the bottom of the page) each encode the current state of one table.

Important SQL files
- page.sql.gz (id, namespace, title, is_redirect, length)
- {page|external|category|...}links.sql.gz links of various types
- user_groups.sql.gz assignment of users to groups (special rights)
Usefull java packages.

Parsing XML documents (Wikipedia history files).
  - Simple API for XML (SAX)\(^1\)
  - Streaming API for XML (StAX)\(^2\)

Reading compressed files (.gz, .bz2, and .7z).

java.util.zip.GZIPInputStream

org.apache.commons.compress.compressors.bzip2.BZip2CompressorInputStream

org.apache.commons.compress.compressors.lzma.LZMACompressorInputStream

\(^1\)https://docs.oracle.com/javase/tutorial/jaxp/sax/
\(^2\)https://docs.oracle.com/javase/tutorial/jaxp/stax/