LABORATORY FOR THE HISTORY
OF SCIENCE AND TECHNOLOGY

EPFL

Semester project Report (2022–2023)

Identifying User Navigation Archetypes In Gallica navigation logs: Sequence Mining and Clustering based approach

Ahmed Nour Achiche, Mohamed Aziz Ben Chaabene

June 15, 2023

Professor:	Jérôme Baudry
Supervisor:	Simon Dumas Primbault
Date:	June 15, 2023

Contents

1 Introduction and problem statement

The contemporary digital era is characterized by an increasing dependency on online resources among researchers across various fields. These online platforms not only serve as repositories for vast quantities of data and knowledge, but they also provide invaluable insights into user behaviors and patterns. This project is primarily centered around documenting and understanding these behaviors on online platforms.

More importantly, the significance of our study extends beyond data analysis to encompass sociological implications. As we analyze and document these digital footprints, we are also exploring how users interact with digital media, and how these interactions shape and are shaped by their social contexts. Thus, the patterns we uncover are not only technological but sociological, shedding light on user preferences, online research strategies, and the social dynamics that underpin these behaviors.

The main objective of our study is to analyze the digital traces that users leave during their navigation of these platforms. These traces are typically captured in server logs and can reveal important patterns and typical behaviors. Analyzing these traces allows us to better understand the preferences of users and the strategies they employ in their online research activities.

Our analysis focuses specifically on the online behavior of researchers using Gallica, a prominent digital library dedicated to preserving French cultural heritage. Gallica, offering a rich assortment of diverse materials, provides an ideal environment for studying online research behaviors.

Through our in-depth analysis, we aim to paint a comprehensive picture of online research behaviors and extract typical user paths. We employ advanced data processing algorithms to identify these paths from the raw server logs. We utilized sequence mining techniques to create embeddings of the sequences, which served as the basis for the subsequent clustering process.

Upon identifying these paths, we strive to group similar user behaviors together using clustering algorithms. This strategy allows us to categorize users into distinct groups, each exemplifying unique online research behaviors.

Our study might also extend to a comparative analysis of the navigation paths of researchers across different research domains, depending on the clustering results. Such an analysis can provide further insights into how different types of researchers interact with online resources, and reveal the influence of the research domain on online research behavior.

This report provides a detailed overview of our research methodology, illustrates our approach to data analysis, presents some preliminary findings, and discusses potential future research directions. It delves into our research process, documenting our challenges and triumphs, and the knowledge we've gained so far. Furthermore, it outlines our plans for further exploration in this intriguing intersection of data analysis and sociology.

2 Gallica Digital Library

Gallica is the digital library of the Bibliothèque Nationale de France (BnF), the National Library of France. It provides free and unrestricted access to a wealth of digitized documents, including books, newspapers, maps, photographs, and sound recordings. These digital resources serve as a rich and readily accessible trove of knowledge for researchers worldwide.

As one of the largest digital libraries globally, Gallica boasts an extensive collection of over 6 million digitized documents available online. It attracts more than one million unique visitors per month, demonstrating its vast reach and influence in the global scholarly community.

Additionally, Gallica is continuously enriched through collaborations with over 1000 partner libraries and institutions. These partnerships contribute to the growth and diversity of Gallica's collection, ensuring that it remains a comprehensive and valuable resource for researchers in various fields.

The digitized documents span a wide range of periods, from the Middle Ages to the 21st century. This extensive temporal coverage allows researchers to trace the evolution of ideas, themes, and trends across centuries, fostering a deeper understanding of their areas of study.

Moreover, Gallica features thematic collections on subjects such as history, literature, science, and culture. These curated collections provide researchers with an organized and focused exploration of specific themes, facilitating efficient and in-depth research.

This diverse collection, available for public access, and its user-friendly interface make Gallica an indispensable resource for researchers. The following sections delve into our methodology for analyzing Gallica's logs and what they reveal about user behaviors and preferences.

3 Data Description

Our analysis draws principally from the comprehensive server logs of Gallica, a rich reservoir of user activity data. The raw data set encompasses a staggering 115GB, documenting over 1 billion HTTP requests recorded throughout the span of 2016 to 2017. Despite the vastness of this data, our study mainly concentrated on a two-month period from February to April. These months were specifically selected for their high activity levels, providing us with a dense and robust data set to refine and validate our methodologies. The goal was to ensure that our developed methods could be confidently scaled and applied to the analysis of longer time periods in the future.

HTTP requests are instrumental in documenting the online behavior of Gallica users. They represent user actions like page requests, file downloads, and link clicks. By analyzing these HTTP requests, we can gain insights into users' interaction with the digital library and understand the patterns and preferences that underpin these interactions.

The provided regular expression pattern, denoted as PATTERN, is utilized for parsing raw data and extracting specific information. The following provides a detailed explanation of how the regex pattern captures the named groups within the pattern.

The pattern is defined as follows:

```
PATTERN = r'^\[(?P<timestamp>.*?)\]\s+"(?P<request_type>\w+)\s+'

    (?P<endpoint>.*?)\s+(?P<http_version>HTTP/\d\.\d)+"\s+

    (?P<status_code>\d+)?\s*(?P<content_length>\d+|\-)?\s+

    "(?P<referrer>.*?)"\s+"(?P<user_agent>.*?)"?\"?\s*(?P<response_time>\d+)?$
```

Now, let's explain how the named groups within the pattern are captured:

- **'timestamp'**: This group captures the timestamp information enclosed within square brackets '[]'. It matches any character (non-greedy) using the '`.*?`' pattern.

- **‘request_type‘**: This group matches a word character sequence representing the type of request, denoted by ‘\w+‘.
- **‘endpoint‘**: The ‘endpoint‘ group captures the endpoint information, which refers to the specific resource being requested. It matches any character (non-greedy) using the ‘.*?‘ pattern.
- **‘http_version‘**: This group matches the HTTP version in the format ‘HTTP/x.x‘, where ‘x.x‘ represents the version number. It specifically matches the pattern ‘HTTP/\d\.\d‘.
- **‘status_code‘** (optional): This group captures the status code of the response, represented by a sequence of digits ‘\d+‘. It is marked as optional with the ‘¿ symbol, allowing for cases where the status code is not present in the data.
- **‘content_length‘** (optional): The ‘content_length‘ group captures the content length information. It can match either a sequence of digits ‘\d+‘ or a hyphen ‘-‘ if the content length is not available. Similar to the ‘status_code‘, it is also marked as optional.
- **‘referrer‘**: This group matches the referrer information enclosed within double quotes ‘""‘. It captures any character (non-greedy) using ‘.*?‘.
- **‘user_agent‘**: The ‘user_agent‘ group captures the user agent information, enclosed within double quotes ‘""‘. It matches any character (non-greedy) using ‘.*?‘.
- **‘response_time‘** (optional): This group captures the response time, represented by a sequence of digits ‘\d+‘. Like the previous groups, it is marked as optional.

By utilizing this regular expression pattern, the named groups are captured and their corresponding information is extracted from the raw data. These captured groups provide valuable insights and enable further analysis and processing of the data.

Here is an example of a typical data entry after being parsed :

- **User Hash:** 4bc3b733c9ad2edfcdf753867b92b5d9
- **Country:** France
- **City:** Valenciennes
- **Timestamp:** 31/Jan/2016:18:59:24 +0100
- **Request Type:** GET
- **Endpoint:** /ark:/12148/bpt6k6105805t/f11.highres
- **HTTP Status Code:** 200
- **Referrer:** `http://gallica.bnf.fr/ark:/12148/bpt6k6105805t/f12.item.r=eisen`
- **User Agent:** Mozilla/5.0 (Windows NT 5.0; rv:31.0) Gecko/20100101 Firefox/31.0
- **Ark:** bpt6k6105805t

In this entry, you can see the HTTP request which indicates that the user retrieved this resource via a 'GET' request. Additional data fields provide context for the request, including the user's country, the timestamp of the request, the HTTP status code, the referring URL, and the user agent, which identifies the browser and operating system used by the user.

Ark (Archival Resource Key)

In the previous example, you can also see an ARK identifier **bpt6k6105805t**.

The digital objects in the Gallica database are identified using Archival Resource Keys (ARKs). These are persistent identifiers assigned to information objects of all kinds, offering a standard, long-term, and globally unique identifier that remains the same even as the actual location of the resource may change over time.

Each ARK is a URL that comprises several components, each providing critical information about the resource. Here is a brief description of the structure:

- **NMA (Name Mapping Authority)** The web address at which the resource can be found. This is replaceable and is not a permanent part of the ARK.
- **ARK Label:** The label that identifies the URL as an ARK, *ark:* in our case.
- **NAAN (Name Assigning Authority Number):** A unique identifier assigned to the institution or entity that has the authority to create and manage the ARK.
- **Name:** The specific identifier that the NAAN assigns. Combined with the NAAN, this creates a globally unique identifier for each ARK.
- **Qualifier:** An optional field that provides additional information about the resource, such as its format, page number, or a specific version of the resource.

A typical ARK structure looks like this:

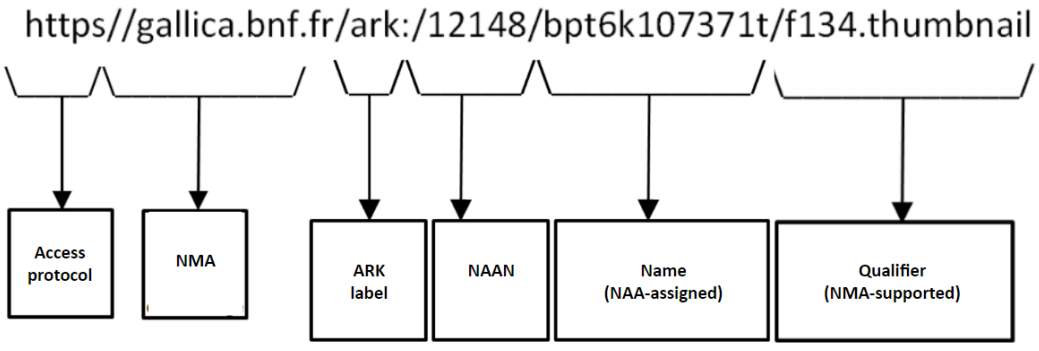

Figure 1: Example of ARK request

In the context of Gallica, ARKs are assigned to the individual digital items, such as books, maps, and photographs, that constitute the library's collection.

4 Literature review

This literature review delves into critical research that has already approached the task of classifying user behavior based on their navigation logs. Furthermore, it provides an in-depth examination of various methods pertinent to the task of our project.

The section can be divided into two main sub-sections, each discussing a different research study:

- **Analysis of Navigation Logs :**
  - Nouvellet's research 1 provides an in-depth exploration of Gallica users' navigation logs.
  - The research focuses on:
    - Description and enrichment of connection logs.
    - Detailed analysis of user paths.
    - Examination of document usage.
    - User provenance and the impact of mediation.
  - Nouvellet's work culminates with a set of recommendations and future perspectives for document clustering and LDA improvements.
- **Interaction with Digital Libraries :**
  - In her doctoral thesis 2, Marwa Trabelsi focuses on studying how users interact with digital library systems through their digital traces.
  - Key aspects of her research include:
    - Emphasis on the extraction of user paths in weakly structured business process systems, specifically digital libraries.
    - Application of process mining techniques to model user interactions and their journey to solve information search tasks.
    - Proposal of a method to identify the paths of users in the digital library Gallica, and a methodology for transforming raw traces into modeled ones.

Following the discussion of each study, we will conduct a thorough examination of these methods. This will allow us to identify their strengths, limitations, and suitability for our project's objectives. We aim to provide the rationale behind our chosen approach through this detailed exploration, thereby confirming its appropriateness and effectiveness within the context of our project.

4.1 Sequential Pattern Mining

Survey [1] provides a crucial overview of Parallel Sequential Pattern Mining (PSPM) methods, an advancement in the data mining field that has emerged as a solution for the inefficiencies of traditional mining methods with large-scale data. The authors have thoroughly surveyed and categorized various PSPM algorithms, highlighting their key ideas, advantages, and disadvantages.

For our project, which focuses on the sequences of user actions in a digital library, this survey offers significant insights. The knowledge about PSPM can potentially assist us in improving the efficiency of our data analysis process by identifying frequent patterns of user behavior. The paper also helps us anticipate challenges we may encounter in implementing PSPM and points out future opportunities in this field. The provided open-source software of PSPM also serves as a practical guide for our algorithm implementation

- Strengths of Sequential Pattern Mining:
  - Ability to detect patterns in sequence data: This is the key strength of SPM. It can identify patterns within sequences of data that other mining algorithms might miss.
  - Efficient algorithms: Algorithms such as PrefixSpan avoid the costly step of candidate generation, making them faster and more memory-efficient than alternative approaches.
  - Adaptable to different domains: SPM can be used in various domains, such as web click-stream analysis, DNA sequence mining, customer purchase behavior, etc.
- Cons of Sequential Pattern Mining:
  - Need for fine-tuning: Algorithms such as PrefixSpan require the setting of a minimum support threshold, which can influence the patterns discovered. If the threshold is set too high, many smaller but potentially interesting patterns might be missed. If set too low, the algorithm could generate an overwhelming number of patterns, including many of less interest.
  - Noise sensitivity: SPM can be sensitive to noise and outliers in the data, which might affect the patterns discovered.
  - Lack of context: SPM identifies patterns based on frequency and order but doesn't necessarily capture the context of the patterns.

This method was used with the state-of-the-art serial sequential pattern mining algorithm PrefixSpan [2]. PrefixSpan, short for Prefix-projected Sequential pattern mining, is an efficient sequential pattern mining method that avoids the costly process of candidate generation and testing. Instead, it mines the frequent prefixes of sequences and projects them into a smaller database to recursively mine the frequent patterns.

The fundamental concepts of PrefixSpan: prefix, projection, and postfix were formalized as the following :

- **Prefix**: Let $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_m\}$ be sequences of sets. B is a prefix of A if and only if the following conditions hold:
  - $b_i \subseteq a_i$ for all $i \in \{1, ..., m\}$, meaning every item in b_i is contained in a_i for every i.
  - $b_m \subseteq a_j$ for some $j \in \{1, ..., n\}$, meaning the last itemset in B is contained in some itemset in A.
  - Every item in b_m is lexicographically (alphabetically) less than or equal to every item in a_j where $b_m \subseteq a_j$.
- **Projection**: Let A, B, and C be sequences with B being a subsequence of A. C is a projection of A w.r.t. (with respect to) prefix B if:
  - B is a prefix of C.
  - There does not exist a sequence C' such that B is a prefix of C', C is a proper subsequence of C' ($C \subset C'$), and C' is a subsequence of A.
- **Postfix**: If $C = \{c_1, c_2, ..., c_k\}$ is a projection of $A = \{a_1, a_2, ..., a_n\}$ w.r.t. prefix $B = \{b_1, b_2, ..., b_m\}$, then $D = \{d_1, d_2, ..., d_p\}$ with $d_i = c_{m+i}$ for $i \in \{1, ..., p\}$, is the postfix of A w.r.t. B, denoted as $D = \text{Postfix}(A, B)$.

If B is not a subsequence of A, then the projection and postfix of A w.r.t. B are defined as empty sets, meaning there's no sequence in A that starts with B.

4.2 Sequence Graph Transform

The Sequential Graph Transform (SGT) is a comprehensive method designed to capture the temporal and structural interaction patterns in sequences. The procedure starts by defining a function $\phi_\kappa(d(l,m))$ to quantify the effect of an event at position l on another event at position m, with κ as a tuning parameter and $d(l,m)$ representing the distance measure. This function is mandated to fulfill certain conditions: it must be strictly greater than 0, strictly decreasing with d, and strictly decreasing with κ. The exponential function is frequently chosen to meet these criteria, defined as

$$\phi_\kappa(d(l,m)) = e^{-\kappa|m-l|}, \forall \kappa > 0, d > 0$$

Following this, the interaction effects are calculated for every possible pair of symbols within a sequence. To manage these interaction effects efficiently, an asymmetric matrix Λ is created, with Λ_{uv} housing all instances of symbol pairs (u,v) where v's position is subsequent to u.

The process then moves on to the definition of the "association" feature, denoted as ψ_{uv}. This feature amalgamates all the effects for each symbol pair and normalizes them. The nature of normalization depends on whether the sequence analysis problem is length-sensitive or length-insensitive. In length-sensitive scenarios, the number of instances of symbol pairs is the normalization factor. Conversely, for length-insensitive situations, the normalization factor becomes the number of symbol pairs divided by the sequence length. This concept is encapsulated mathematically as,

$$\psi_{uv}(s) = \begin{cases} \frac{\sum_{\forall(l,m)\in\Lambda_{uv}(s)} e^{-\kappa|m-l|}}{|\Lambda_{uv}(s)|} & ; \text{length sensitive} \\ \frac{\sum_{\forall(l,m)\in\Lambda_{uv}(s)} e^{-\kappa|m-l|}}{|\Lambda_{uv}(s)|/L^{(s)}} & ; \text{length insensitive} \end{cases}$$

Finally, these psi_{uv} features are aggregated to compose a feature representation, denoted as $\Psi(s)$, of the entire sequence.

The SGT method offers several advantages for generating embeddings for user logs action sequences:

- **1. The capture of Temporal Patterns:** SGT succeeds in capturing the relative positions and orders of events within a sequence, a feature that is crucial when analyzing user logs, where the sequence of user actions is very significant.
- **2. Interpretability:** Every individual entry within the SGT feature vector corresponds to a distinct pair of symbols, adding interpretability and delivering insightful information for studying user actions.
- **3. Flexibility:** The SGT method incorporates a tuning parameter κ, enabling control over the influence of distant events on each other and providing adaptability to diverse user behaviors.
- **4. Applicability to variable-length sequences:** SGT is capable of handling sequences of varying lengths, a characteristic frequently observed within user logs.
- **5. Scalability:** SGT's capability of transforming sequences into fixed-length feature vectors facilitates its seamless integration with an extensive range of machine-learning algorithms.

5 Methodology

5.1 Data processing

Preprocessing of logged requests

The initial server logs from Gallica contained a significant amount of data that was either not relevant to our research objectives or not suitable for direct analysis. Therefore, the first step in our data analysis process involved filtering and preprocessing the raw data.

- **Feature Extraction:** We began by extracting relevant characteristics from each HTTP request, such as the timestamp, request type, endpoint, and user agent. This allowed us to isolate the most important information for each request.
- **Request Classification:** The preprocessing step involved classifying the HTTP requests based on their attributes and filtering out those that were not deemed relevant for the analysis. We developed different categories and assigned a binary value to each one, where 1 signifies the presence of the attribute, and 0 its absence. You can find in the appendix the file tree containing the main directories of URL requests with the highlighted directories being the ones we are interested in , you can find as well how the categories we created are broken down in Table **??**.

  After this preprocessing step, only the actions that provided the most useful information for the study were kept for further analysis. These were the homepage, document, blog, simple search, advanced search, filtering search results, page download, document download, pagination, heading, mode, and zoom actions. All other actions were filtered out in the dataset to focus on the most meaningful user interactions. This selective focus streamlines the data for subsequent stages of analysis. This classification step allowed us to focus on the most meaningful user actions and to disregard non-essential requests, resulting in a reduction in data size by approximately 65%.
- **ARK Request Qualifier Extraction:** For ARK requests, we extracted the qualifiers that provided further insights into the user's interaction with the resource. This included information like the viewed page, the reading mode used, and the document's language.
- **Filtering bot Requests:**

  To maintain the integrity and relevance of the data, it was imperative to filter out bot-generated traffic. For this purpose, a user agent parsing library was employed. This ensured that the analysis and embeddings generated were more representative of human user behavior.
- **Filtering Non-Essential Document Requests:** In the analysis of the parameters attached to document requests, we gained insight into the user interaction with Gallica's content. Not all of these parameters, however, are meaningful for our current analysis. For instance, parameters associated with the display of thumbnail images, which do not significantly impact the user's browsing or reading experience, are not central to this study and can be disregarded.

Here's a breakdown of the most frequently occurring document parameters that weren't relevant to our analysis:

- **thumbnail** : Refers to the small version of an image or page.
- **highres:** Indicates a high-resolution version of a document or image, This is generally generated when we execute a page change and for each page change, two of these requests are observed .

- **r :** The exact meaning of this parameter is not immediately clear without additional context.
- **hl :** Referring to highlighted sections or elements.
- **lowres :** Indicates a low-resolution version of a document or image.

To refine our analysis, we've decided to focus on the document parameters that represent significant user interactions. These are:

- **item:** Relates to a particular item or object within a collection.
- **zoom:** Indicates that a user has zoomed in or out on a document or image.
- **planchecontact:** Could refer to a specific viewing mode in Gallica, like a contact sheet.
- **vertical and double:** These refer to different viewing modes or layouts (e.g., double-page view).
- **pagination:** Refers to the navigation through different pages of a document.

By filtering out the non-essential parameters, we reduced the number of document requests by 85%. This refinement allows us to focus on the most relevant user interactions and provides a cleaner, more manageable dataset for the subsequent stages of analysis.

It is important to note, that some data preprocessing challenges arose due to the occasional randomness in the data format and difficulty in differentiating certain types of requests, such as page and section requests. Moreover, some document qualifiers were unpredictable. For instance, an ARK request might have a qualifier like **ark:/12148/bpt6k76400z.chemindefer**, which is not immediately recognizable.

To address these challenges, we leveraged the navigator developer tool to simulate actions on Gallica and observe the requests generated to pinpoint the actions we should keep for our analysis.

Scaling

Given the vast magnitude of the dataset at hand and the necessity for a scalable preprocessing pipeline, it was imperative for us to perform request preprocessing in smaller chunks. This approach allowed us to process the data in a more manageable manner. However, it is important to note that this chunking process may result in the segmentation of sessions at the boundaries, thereby potentially splitting sessions that would otherwise be considered as a single cohesive session.

Sessionization

The sessionization process in our analysis involves several steps that help us group related user interactions (or "sessions") together, filter out less relevant data, and calculate some basic statistics for each session. Here is an explanation of each step in our sessionization pipeline:

- **Group the DataFrame by User's IP Address and Timestamp:** We start by grouping our data by the user's IP address and the timestamp of each request. This helps us track the sequence of actions performed by each user in chronological order.

- **Filter User Requests with Less Than 5 Total Requests:** We eliminate noise from the data by discarding the users who made fewer than 5 total requests. This step reduces the likelihood of including irrelevant or accidental interactions in our analysis.
- **Calculate Time Difference Between Consecutive Log Entries for Each User:** To determine the duration of each user's interactions, we calculate the time difference between consecutive log entries for each user.
- **Set a Threshold to Determine the Start of a New Session (60 minutes):** We establish a 60-minute threshold of inactivity to separate individual sessions. If the time difference between two consecutive log entries exceeds 60 minutes, we assume that the user has started a new session.
- **Determine the Start of Each Session:** Using the 60-minute threshold, we identify the start of each session.
- **Calculate Request Frequencies for Each Session:** For each identified session, we calculate the frequency of requests. This gives us a measure of user activity during that session.
- **Filter Sessions with More Than 1 Request/Second:** To further refine our data, we filter out sessions with an average request frequency of more than one per second. This step helps to exclude possible bot activity, which is not relevant to our analysis of human user behavior.
- **Generate a Session ID:** Finally, we generate a unique ID for each session, using the format S_FileNumber_SessionNumber_U_HashedUserIPAddress where the file number refers to the number of the current preprocessed data chunk being passed through the sessionization pipeline. This ID allows us to reference specific sessions in our subsequent analysis.

After the sessionization process, a second processing step was implemented to derive interpretable actions that describe the sequential user behavior within each session. This additional step aimed to provide a more comprehensive understanding of user interactions. The resulting actions were documented in Figure 2 . However, we found that this granularity wasn't optimal for sequence classification (explored further below) and thus chose to apply our methods to the highlighted actions in the figure. You can also find in Table 3 of the appendix more details about each action, allowing for further analysis and interpretation.

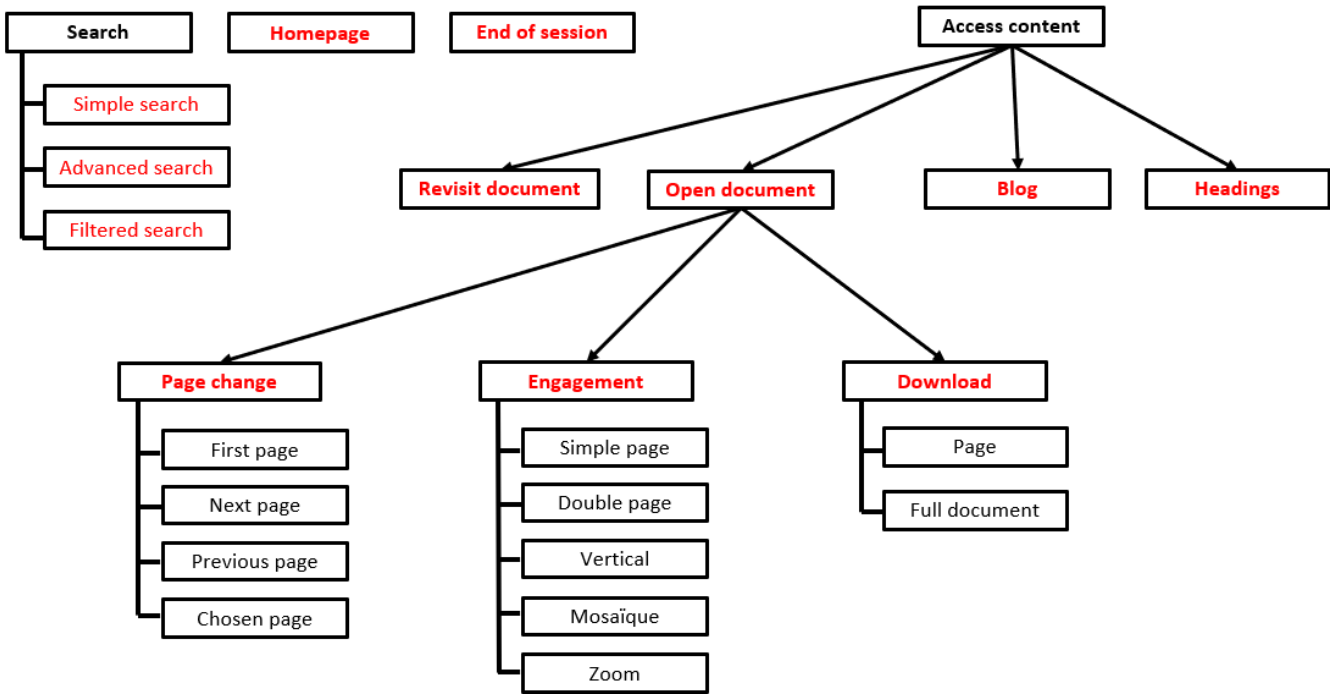

Figure 2: Action tree

In Figure 3, we present the number of sessions per day observed between January 31, 2016, and April 3, 2016. The curve demonstrates notable fluctuations throughout the analyzed period. On or around March 10th, a peak of approximately 25,000 sessions is observed. However, following this peak, there is a significant and rapid decline in the number of sessions, which stabilizes at around 4,000 sessions until the end of the covered timeframe. The total number of sessions recorded during this entire period amounts to 622958 sessions.

Figure 4 displays the histogram representing the distribution of the number of actions per session over the same period. Based on the histogram, the data exhibits a skewed normal distribution, characterized by positive skewness. The majority of sessions, ranging from the 5th percentile to the 95th percentile, have between 4 and 230 actions per session. The mean session length is calculated to be 67, while the median session length is 22. This indicates that some sessions have exceptionally long durations compared to others. This observation is further supported by the analysis of the maximum number of actions per session, which is recorded as 37 415.

To mitigate the impact of these abnormally long sessions in several of our applications, we have chosen to focus on sessions that fall within the 0.05 and 0.95 percentiles, thereby excluding the extreme outliers from our analysis.

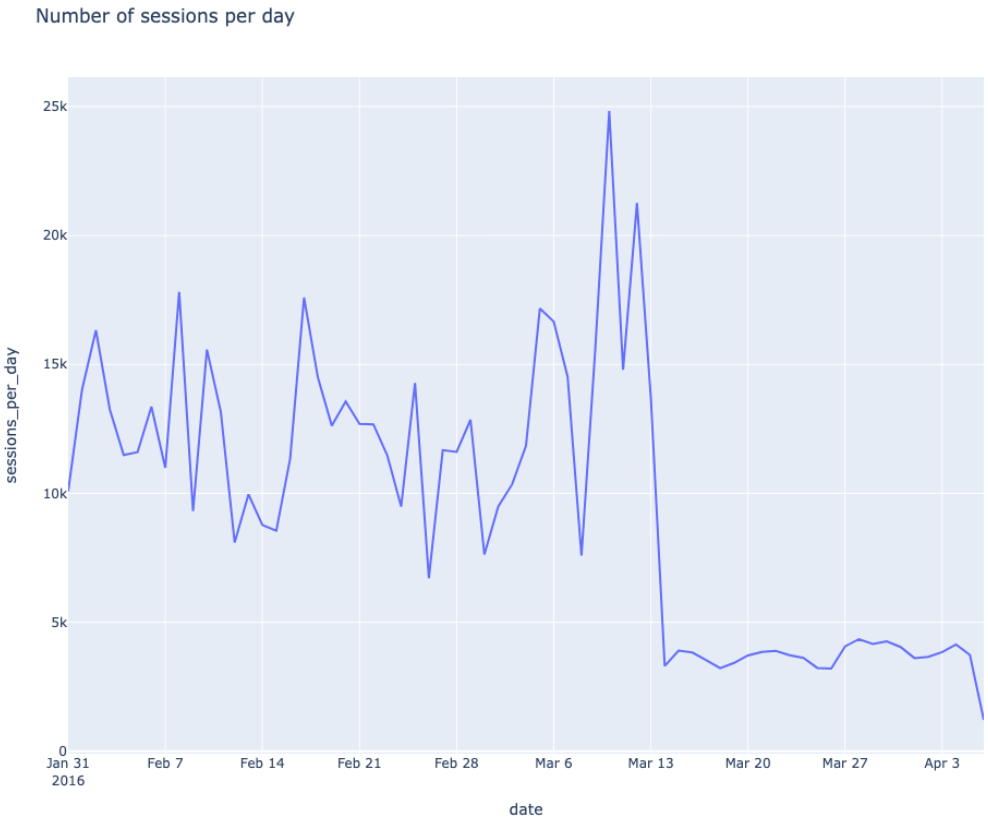

Figure 3: Number of sessions per day (2 months)

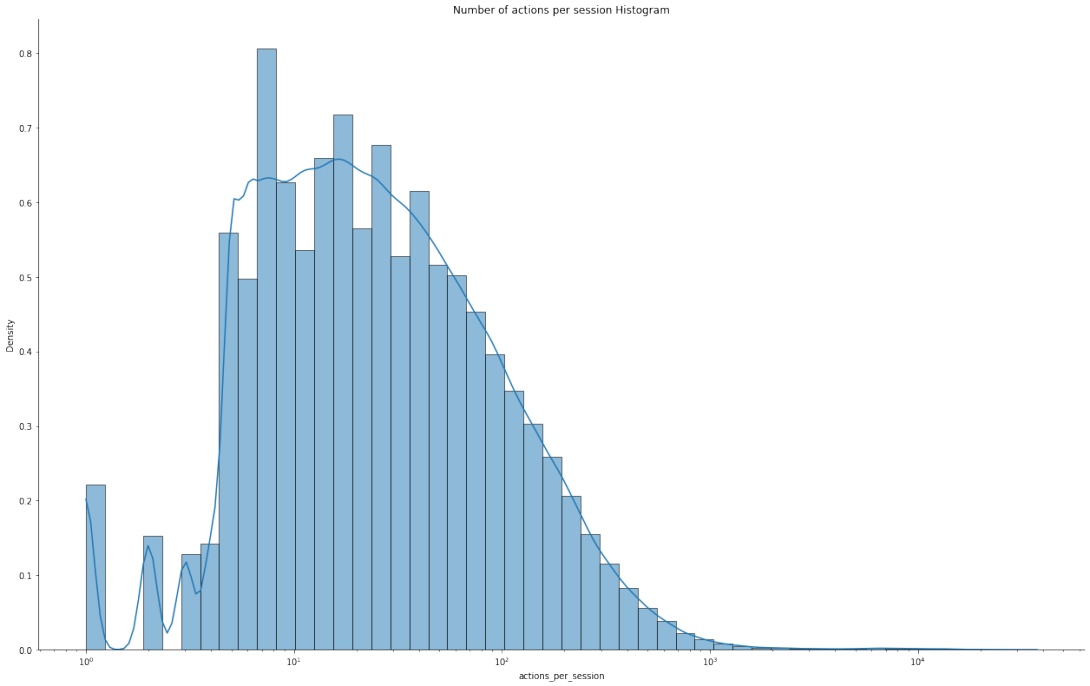

Figure 4: Number of actions per session histogram (2 months)

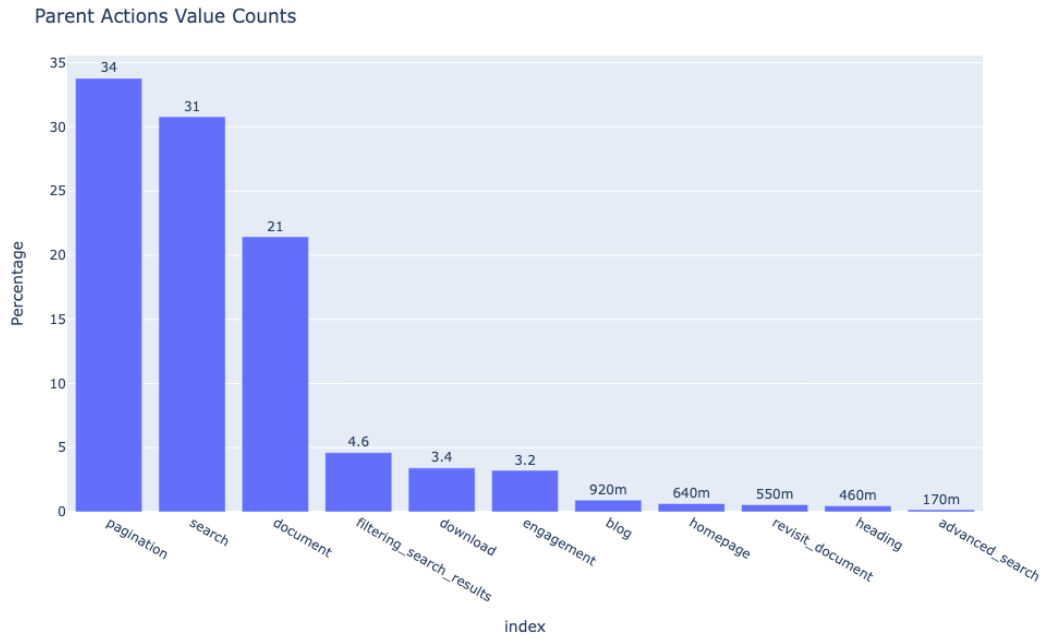

Figure 5: Distibution of action types (2 months)

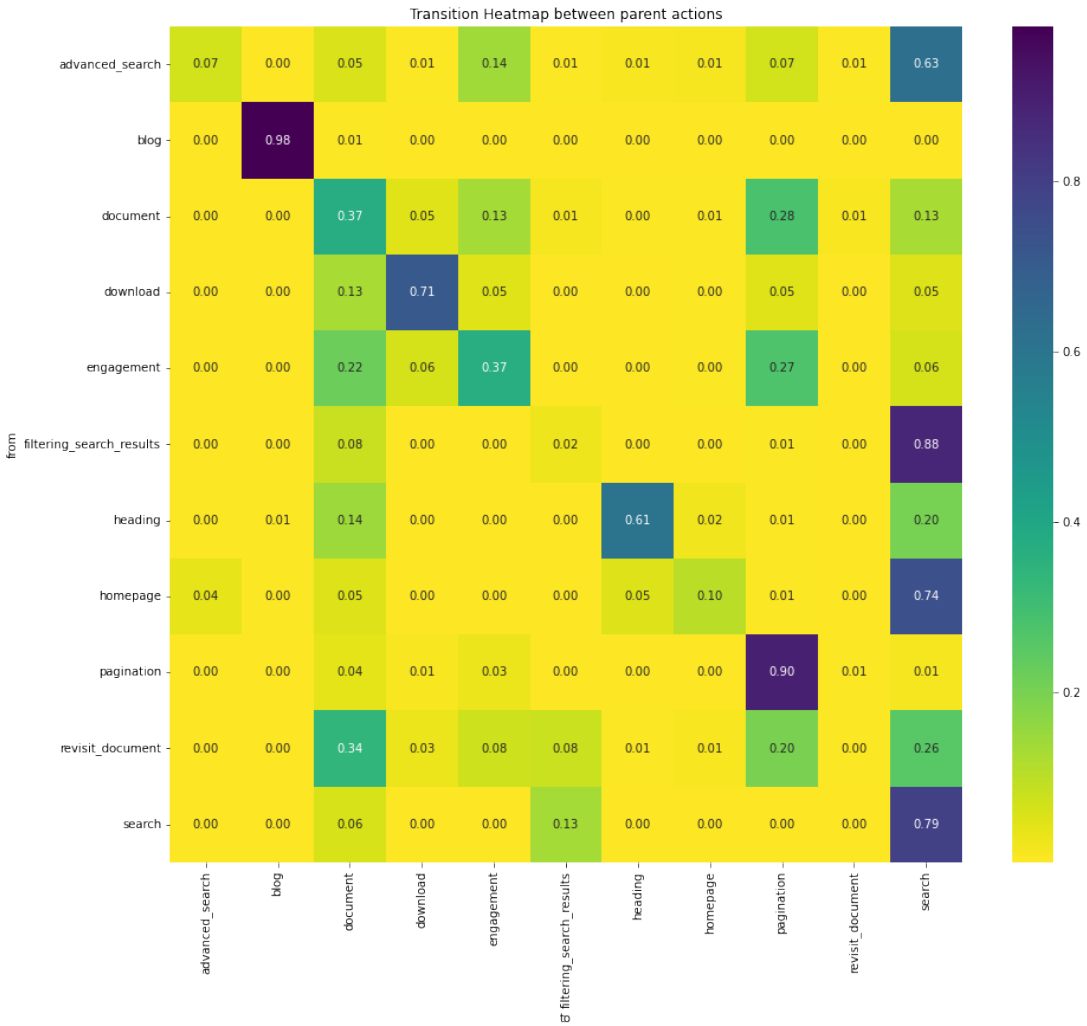

Figure 6: Transition matrix between actions Normalised by line (lines sum to 1) (2 months)

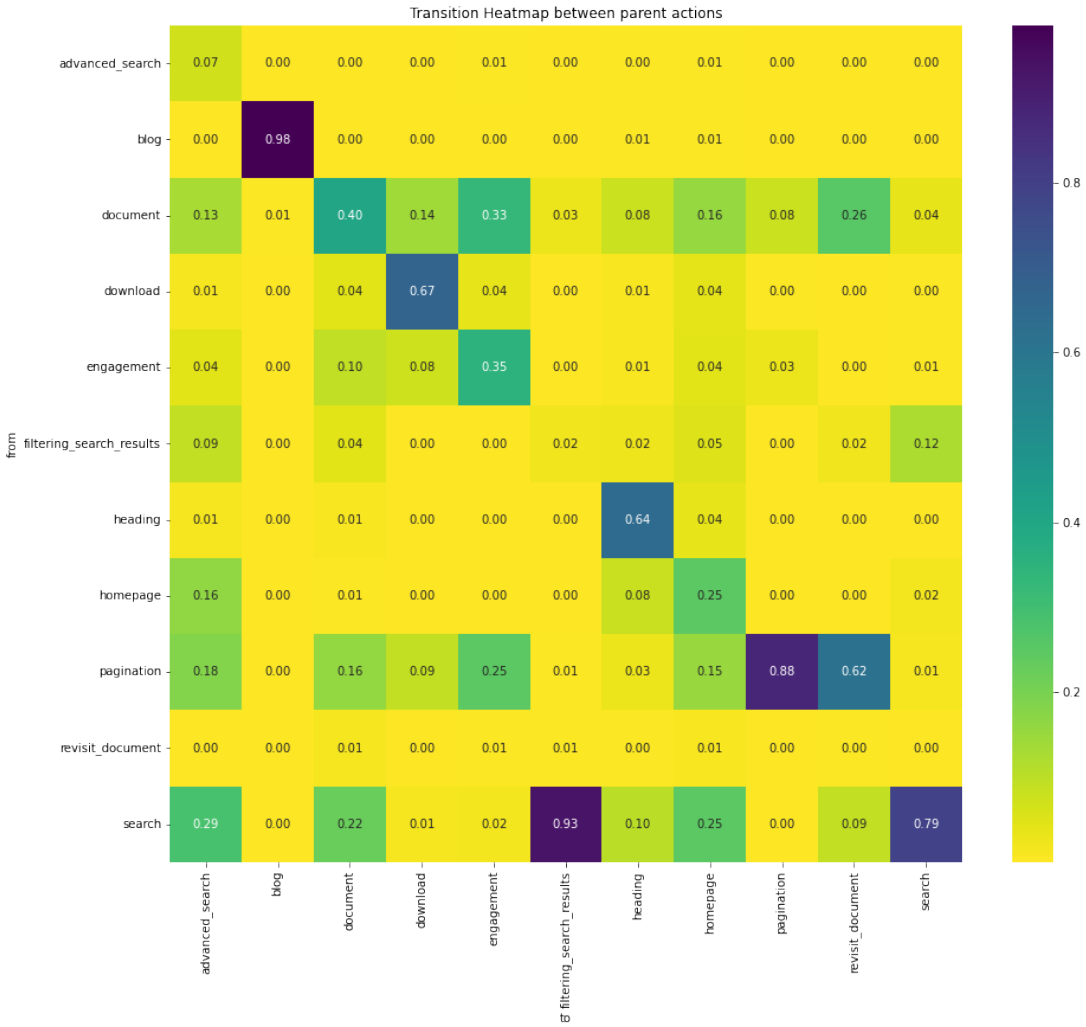

Figure 7: Transition matrix between actions Normalised by columns (columns sum to 1) (2 months)

Figure 6

- Users exhibit a tendency to rapidly open documents in succession, as evident from the document-to-document transition coefficient of 0.41.
- Users who engage with actions while browsing a document are more likely to continue their engagement by performing subsequent actions indicating involvement (transition coefficient of 0.39). Additionally, they are more inclined to navigate to another page within the same document rather than opening a different document (transition coefficient of 0.27 compared to 0.22, respectively).
- A significant majority of users who visit the homepage proceed to conduct a simple search as their next action.
- A considerable proportion of searches conducted by users result in an unsuccessful outcome. Subsequently, users are inclined to perform another search in 79

Figure 7

- An advanced search is most commonly preceded by a simple search.
- Downloads are most frequently preceded by another download action, accessing a document, or navigating through pagination, respectively.
- Visiting the homepage is most likely preceded by an unsuccessful search, indicating users' probable intention to modify their search criteria or approach.
- Revisiting a document is commonly observed following pagination through another document.

In Figures 6 and 7, the depicted transition matrices offer valuable insights into the overall behavior of Gallica users. In Figure 5, several notable patterns can be observed. Users demonstrate a tendency to rapidly open documents in succession, indicating an active exploration of the available resources. Engaged users, who perform actions indicating involvement while browsing a document, are more likely to continue their engagement by performing subsequent actions that signify their continued interest. Additionally, users navigating within a document are more inclined to proceed to another page within the same document rather than opening a different document altogether. Furthermore, the majority of users visiting the homepage proceed to conduct a simple search as their next action. It is worth noting that a significant proportion of searches conducted by users yield unsuccessful results, prompting users to perform follow-up searches, apply filters to refine search results or open a document directly.

Figure 7 provides further insights into the sequential behavior of Gallica users. Noteworthy observations include the fact that an advanced search is most commonly preceded by a simple search, indicating users' progression towards more refined queries. Downloads are frequently preceded by another download action, accessing a document, or navigating through pagination, suggesting users' recurrent interest in obtaining multiple resources. Visiting the homepage often follows an unsuccessful search, indicating users' inclination to revise their search criteria or approach. Additionally, revisiting a document is frequently observed following pagination through another document.

5.2 Session embeddings

In this section, our primary aim was to analyze sequential data resulting from the sessionization process. We attempted to extract meaningful patterns from each session's sequences of actions, employing

three methods: Long Short-Term Memory Networks (LSTM), Sequence Graph Transform (SGT), and Sequential Pattern Mining (SPM).

Initial attempts at using LSTM, while theoretically promising, failed to yield satisfactory results in our context. As a next step, we experimented with SGT, which provided fruitful insights. The third method, SPM, also centered around pattern-based embeddings, aligning with our pursuit to capture underlying sequence patterns.

In addition to these algorithmic techniques, we also constructed custom features, designed to represent distinct behavioral patterns we expect to see in the data.

The following sections will provide more detailed information about our feature extraction process, as well as the LSTM, SGT, and SPM methods.

Hancrafted features

For each of these sessions, we calculated various metrics that serve as our features for modeling:

- **Session length**: This is simply the total count of actions in a session, providing a measure of the session's overall length. It gives us a sense of how engaged a user is within a single session, with longer sessions potentially indicating higher engagement or complexity of tasks being performed.
- **session_duration**: This feature would typically represent the total time span of a session, from the user's first action to their last. Like session length, this feature can give us a sense of user engagement, but on a temporal scale rather than based on action count.
- **Unique actions**: This feature calculates the number of distinct actions within a session. A higher count might indicate a user who is exploring different functionalities, whereas a lower count could suggest a more focused or repetitive use of the platform.
- **Search count**: This represents the number of 'search' actions performed in a session. It gives us an idea about how much a user is exploring or seeking specific content within a session.
- **Number of unique documents accessed**: This feature is the count of 'document_access' actions in a session. It signifies how many documents the user has accessed, showing how much content they're engaging with.
- **Search count to documents accessed count ratio**: This ratio provides insight into the balance between search and content engagement. If this ratio is high, it might indicate a user who is performing more exploratory searches than accessing actual documents, or perhaps having difficulty finding what they're looking for.
- **Average number of actions before opening a new document**: This feature calculates the average number of actions performed between each 'document_access' action. It provides insight into the user's behavior between accessing documents. For instance, a higher number might suggest a user who is taking time to explore or perform other tasks between accessing documents.
- **Average time between actions**: This feature calculates the average time interval between consecutive actions within a session. It can help identify the pace at which a user interacts with the system and estimate engagement.

LSTM

In our exploration of suitable methods for the task at hand, We briefly attempted to utilize Long Short-Term Memory (LSTM), a recurrent neural network, to process our sequences of user actions. The process involved several steps:

- Data Preprocessing: User actions were converted into integer sequences, padded to ensure consistent length, and normalized to a 0-1 range.
- Reshaping: Sequences were reshaped to match the LSTM input format of (samples, time steps, features).
- Model Construction:
  - Input Layer: The sequences as input.
  - LSTM Layer 1 and Dropout Layer 1: First LSTM and its dropout layer.
  - LSTM Layer 2 and Dropout Layer 2: Second LSTM and its dropout layer.
  - Dense Layers: Additional layers to facilitate complex representations.
  - Output Layer: The encoded representation of the sequence.
- Training: The model was trained on the reshaped sequences, applying early stopping to avoid overfitting. Best weights were restored when loss improvement halted.
- Encoder Extraction: The encoder, a lower-dimensional representation of our sequences preserving their temporal dynamics, was extracted from the autoencoder.

Despite our best efforts, the high sensitivity of LSTM to sequence length rendered it an unsuitable choice for our varied data. This hurdle compelled us to explore alternatives and adjustments, such as limiting the initial sequence length to a defined range, for instance the first 10 to 20 sequences. Another adaptation involved decreasing the granularity of the actions, thereby reducing the size of the action vocabulary. These revisions were pursued with the aim of accommodating the LSTM model to our data specifics and requirements.

SPM

Sequential Pattern Mining (SPM) is a data mining technique aimed at discovering or identifying regular patterns or sequences of items in transactional data or between sets of items in a given dataset. SPM techniques are highly useful when analyzing data where the sequence of occurrences is important, such as user interactions in a web session.

SPM has been successfully used in a myriad of applications to navigate complex and large volumes of data, often collected from shared resources. The challenges it overcomes include excessive memory cost, slow processing speed, and insufficient hard disk space.

As part of this project, Sequential Pattern Mining (SPM) emerged as a viable method to address the issues experienced with conventional data mining approaches and was applied to 60 days of data. Our goal is to identify typical behaviors from the refined sequences of actions of each session. Applying SPM would yield fundamental patterns present in these sequences, and according to the presence or not of these patterns, one could classify them.

To extract these embeddings, we used one of the most popular algorithms for SPM is PrefixSpan (short for "Prefix-projected Sequential pattern mining"). PrefixSpan is known for its efficiency as it avoids the

high computational cost of candidate generation and testing. The algorithm operates by taking each frequent prefix as a projected database and mines its frequent patterns recursively.

This algorithm takes as input the dataset of sequences and an F, the minimum number of occurrences of patterns to consider them as fundamentals patterns.

The determination of the optimal 'F' value, which represents the minimum number of pattern occurrences to be considered fundamental, can be challenging due to its direct link to the size of the dataset and the complexity of the patterns. Setting 'F' to a high value means that only patterns appearing very frequently in the data will be deemed significant. Conversely, a lower value of 'F' increases the risk of including patterns that may have occurred by chance and are not truly representative of the underlying behavior.

To provide a simple example of how Sequential Pattern Mining (SPM) would work, let's consider a set of sequences displayed in Table 1. In this case, the sequences represent user actions, where 'a', 'b', and 'c' could represent different types of actions such as 'search', 'document access', 'page change', etc.

Table 1: Example of sequences of user actions and the patterns identified using PrefixSpan with minimum support of F=2.

Sequence of Actions	**Patterns (F=2)**
a → b → a → a → b	(a) , (b) , (a → b), (b → a), (a → b → a), (b → a → a)
a → b → c	(a) , (b) , (c) , (a → b)
a → b → a	(a) , (b) , (a → b), (a → b → a)
b → a → b	(a) , (b) , (b → a)
b → a → c	(a) , (b) , (c) , (b → a)
b → a → a → b	(a) , (b) , (b → a), (b → a → a)

As we can see, patterns of length 1 and 2 seem to be very prevalent and might not be the best choice of embedding to differentiate between sessions, thus , we only considered patterns with length ≥ 3

In our case, three different 'F' values were tested: 100k, 120k, and 150k. These produced considerably different results. This choice was made taking into account that we have in total 592000 sessions on the 60 days period. At 100k, the algorithm identified a vast number of patterns 699, but also takes a fairly long run-time, and going below this threshold seems unnecessary. When 'F' was increased to 120k, the number of fundamental patterns decreased dramatically to 255. A further increase to 86 led to an even more drastic reduction, leaving us with only 35 patterns.

The substantial variation in the number of patterns obtained with different 'F' values emphasizes the sensitivity of the SPM to this parameter. Selecting the appropriate 'F' is a balancing act between obtaining a manageable number of meaningful patterns and not overlooking less common, but potentially important, patterns. This process typically requires multiple iterations and a deep understanding of the data and the context in which it is applied.

Given the circumstances, we decided on 'F' = 100k. This value provided a comprehensive list of 670 patterns after filtering, offering a detailed snapshot of user behavior within the sessions. While this choice results in a longer runtime, we deemed this an acceptable trade-off for the rich behavioral insights it generates.

Once the fundamental patterns were identified, we mapped these to the corresponding sessions. This was accomplished by creating a binary representation for each session, where a '1' signifies the presence of a pattern, and a '0' indicates its absence. This conversion allows for a numeric representation of the data, facilitating subsequent stages of our analysis such as dimensionality reduction and visualization.

Following this, we utilized Truncated Singular Value Decomposition (Truncated SVD) to reduce the dimensionality of our binary pattern matrix. By testing a range of component numbers which can be seen in Figure 8 , we observed that three components could capture approximately 90% of the variance in the data, thus, we proceeded with these three components. Subsequently, we applied UMAP for

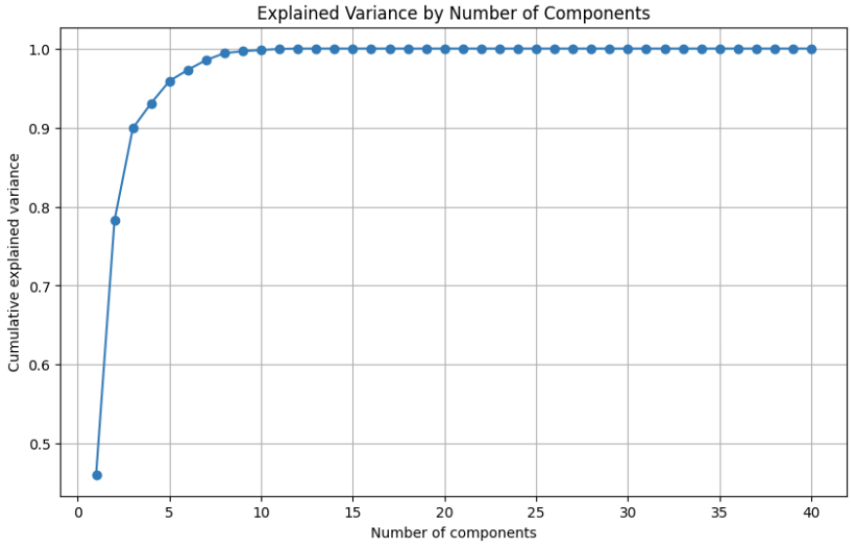

Figure 8

dimensionality reduction, directly on the binary pattern matrix. Unlike the previous stage where we employed Truncated SVD, we directly set the number of UMAP components to three. This was in line with our observation from the explained variance plot. Moreover, by setting the metric to 'cosine', we ensured that the distances in the reduced space corresponded to angular distances in the original high-dimensional space, making the embeddings more suitable for visualization and interpretation.

Finally, the UMAP embeddings were visualized in Figure 9 to provide a three-dimensional projection of the sessions. This visualization highlighted potential clusters and offered insights into common behavioral patterns across the sessions. To summarize, our methodology leverages sequential pattern mining, dimensionality reduction, and visualization techniques to identify, represent, and interpret common behaviors in user sessions. This combination of techniques effectively distills complex sequential data into meaningful insights, contributing to our understanding of user behavior and allowing for an optimized design of web sessions based on these findings.

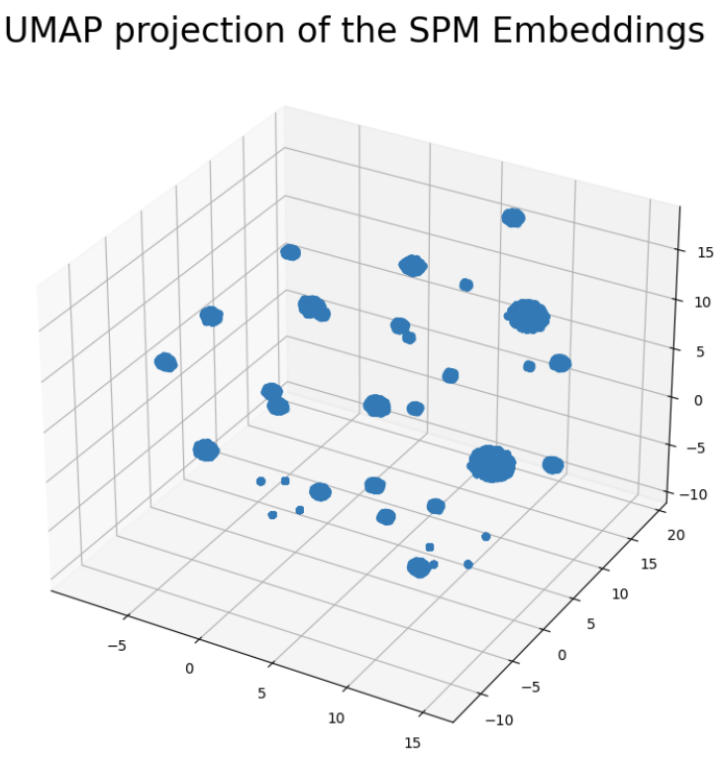

Figure 9

SGT

In the SGT method, a key parameter to consider is kappa, which plays a crucial role in determining the sensitivity to short-term or long-term patterns during the feature construction process. Initially, our approach involved implementing a grid search technique with kappa and the number of clusters as hyperparameters. We aimed to identify the optimal combination that would yield improved clustering results. To evaluate the quality of the clusters, we employed two metrics: silhouette scores and the db_index.

Regrettably, when applying the supposedly optimal parameters to sessions consisting of two days' worth of data, the resulting clusters were not easily interpretable. To address this issue, we adopted a visual inspection method wherein we displayed the ten closest sessions for each centroid. Through this visual examination, we assessed which number of clusters produced the most interpretable outcomes.

Similarly, when dealing with a larger dataset spanning two months, we encountered challenges in obtaining interpretable results compared to the outcomes generated by the spm method. The obtained results were notably less comprehensible and did not provide a clear understanding of the underlying patterns within the data.

5.3 Clustering

Dimensionality reduction: Uniform Manifold Approximation and Projection

UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique employed to visualize and analyze high-dimensional data. It aims to capture the inherent structure and relationships within the data while preserving both local and global patterns. UMAP excels at preserving the intrinsic geometry of the data, motivating its choice over other techniques like t-SNE that do not do equally well in preserving data structure.

Let us denote the high-dimensional input data matrix as **X**, where each row represents a data point and each column corresponds to a feature. UMAP constructs a low-dimensional representation **Y** of the data, typically in two or three dimensions, enabling effective visualization.

The UMAP algorithm encompasses the following fundamental steps:

- **1. Construction of a neighborhood graph:** Initially, UMAP constructs a weighted neighborhood graph through the nearest neighbor search. For every data point, the algorithm identifies its nearest neighbors based on a chosen distance metric, such as Euclidean distance or cosine similarity. The strength of the connections between data points is determined by the distances to their respective neighbors. This process captures the local structure of the data.
- **2. Optimization of the graph representation:** UMAP optimizes the graph representation by minimizing the discrepancy between pairwise similarities of the data points in the high-dimensional space **X** and the low-dimensional space **Y**. This optimization is performed using a stochastic gradient descent-based approach. The objective is to discover a low-dimensional representation that preserves the local structure encoded in the neighborhood graph.
- **3. Construction of fuzzy set membership:** UMAP establishes a fuzzy set membership for each data point by interpolating between its neighboring points in the graph. This fuzzy set membership encapsulates the notion of connectivity and facilitates a smooth transition between adjacent data points.
- **4. Embedding of the fuzzy set into the low-dimensional space:** Finally, UMAP embeds the fuzzy set membership into the low-dimensional space **Y** utilizing a mathematical technique known as fuzzy simplicial set representation. This process ensures the preservation of both local and global structures of the data, thereby providing an effective dimensionality reduction.

Kmeans

The k-means algorithm is a widely used clustering technique that aims to partition a given dataset into k distinct clusters. It operates based on the principle of minimizing the within-cluster variance or sum of squared distances from data points to their respective cluster centers. The algorithm seeks to find the best representation of each cluster center and assigns data points to the cluster with the closest center.

Let us denote the input dataset as **X**, consisting of n data points in a d-dimensional space. The k-means algorithm comprises the following key steps:

- **1. Initialization:** The algorithm begins by randomly selecting k initial cluster centers. These centers can be chosen randomly from the data points or using other initialization methods such as k-means++.
- **2. Assignment of data points:** Each data point is assigned to the cluster with the nearest center. The distance metric used is typically the Euclidean distance, although other distance measures can be employed depending on the data and problem domain.
- **3. Update of cluster centers:** After assigning data points to clusters, the algorithm calculates the new centroid (center) for each cluster by computing the mean of the data points assigned to that cluster.
- **4. Iteration:** Steps 2 and 3 are repeated iteratively until convergence. Convergence is achieved when the cluster assignments no longer change or when a predefined maximum number of iterations is reached.

The k-means algorithm optimizes the clustering by minimizing the objective function, known as the within-cluster sum of squares (WCSS):

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2$$

where C_i represents the data points assigned to cluster i, and $\mathbf{c}_i$ denotes the centroid of cluster i.

A significant challenge with the k-means algorithm is determining the appropriate number of clusters (k). A sub-optimal choice for k can lead to ineffective clustering. To overcome this, we utilized the Elbow Method, a heuristic used in identifying the optimal number of clusters in a dataset. This method consists of:

- **1. Compute k-means clustering:** Perform k-means clustering and compute the WCSS for a range of k values.
- **2. Plot WCSS:** Plot the WCSS against the range of k values.
- **3. Identify the elbow point:** The "elbow" point is identified as the point where the rate of decrease in WCSS sharply shifts. This is visually determined as the point where the plot starts descending much more slowly.

Though the Elbow Method provides a visually informed estimate of the best number of clusters, it should be noted that it remains somewhat subjective, as the "elbow" may not always be clear or well-defined.

With the number of clusters determined via the Elbow Method, we proceed with our proposed methodology.

The k-means algorithm iteratively updates the cluster assignments and recalculates the cluster centers until convergence. The resulting clusters represent compact and coherent groups of data points.

In our proposed methodology, we adopt a two-step approach to identify relevant clusters within the data. First, we employ dimensionality reduction techniques using UMAP on the session embeddings. This step allows us to reduce the dimensionality of the data while preserving the underlying structure and relationships among the sessions.

Once we have obtained the reduced-dimensional session embeddings, we proceed to apply the k-means algorithm. The session embeddings serve as input to the k-means algorithm, which is responsible for clustering the sessions into distinct groups.

The combination of UMAP for dimensionality reduction and k-means for clustering provides a robust approach to uncovering meaningful patterns and grouping similar sessions together. UMAP reduces the complexity of the data, allowing for a more efficient clustering process by k-means. This methodology enables us to explore and interpret the data in a more manageable and insightful manner.

6 Results

Preliminary Results (spanning 2 days of logs)

SGT

This subsection presents the results obtained from applying the Sequence Graph Transformation (SGT) method to two days' worth of data. The clusters generated by the SGT method are analyzed, and their patterns and interpretability are discussed. Interactive HTML graphs and an image of the clustering results can be accessed at the following link. Additionally, the resulting images are added to the appendix (14 to 23).

The following is a description of the ten sequences closest to the ten centroids of the clusters found using the SGT method:

- **Cluster 1**: This cluster is primarily composed of accessing a document followed by long sessions of pagination, occasionally accompanied by intermittent engagement or search activities.
- **Cluster 2**: The most notable pattern in this cluster is starting with the homepage, followed by a long search, accessing two or more documents, and concluding with a download. The initial access to the homepage is particularly interesting, suggesting that the user is familiar with Gallica and is searching for a specific document to download. The occasional presence of search result filtering and engagement further supports this idea.
- **Cluster 3**: The most remarkable feature of this cluster is the consistent pattern of accessing a document, followed by engagement and a long search, accessing another document, and concluding with extensive pagination. This behavior characterizes users who are seeking specific information. The occasional pattern of accessing a document, engaging, returning to the homepage, and conducting a long search suggests that the user was unsatisfied with the information found in the initial document.
- **Cluster 4**: The most frequent pattern in this cluster appears to be searched, accessing a document, extensive pagination, and finally, a download. This pattern closely resembles Cluster 3 but includes the addition of downloading the desired information. Once again, this behavior suggests that users are seeking specific information that they intend to retain, such as an image or a page from a document.
- **Cluster 5**: Sessions in this cluster are short and typically consist of accessing a document, brief pagination, and session abandonment. This pattern indicates that the user is a casual surfer who may have accessed a Gallica document from an external source, briefly viewed a few pages, and then continued browsing elsewhere.
- **Cluster 6**: It is challenging to infer a regular pattern from this cluster, except for the occasional presence of advanced search, which was absent in the previously examined clusters. The 2D graph also indicates that this cluster is not very compact, which partly explains the irregularity of the discovered patterns.
- **Cluster 7**: Typically, one or more accesses to a document are followed by pagination and/or engagement, and eventually, a download. This cluster bears a resemblance to Cluster 4, as reflected in the 2D graph.
- **Cluster 8**: This cluster seems to reflect a behavior discussed earlier, which involves accessing multiple documents and extensive pagination. However, contrary to expectations, it is not closely located in Cluster 5 in the feature space. The implications of this deviation are uncertain.

- **Cluster 9**: This cluster is not interpretable in the context of our task, but it appears to reflect the internal mechanics of the Sequence Graph Transformation (SGT) algorithm. The 2D graph clearly shows that this cluster is densely populated, and its formation seems to be based solely on the consistent presence of a fixed-length sub-pattern (a single document access followed by three paginations).
- **Cluster 10**: Characterized by short and goal-oriented sessions, this cluster follows a pattern of accessing a single document, brief pagination, and multiple downloads.

LSTM

When we applied the LSTM method directly to the sequences, we discovered that it tended to group the sequences primarily based on length. This phenomenon can be attributed to the LSTM's inherent sensitivity to sequence length. In effect, this resulted in an overemphasis on the number of actions in a session, potentially overshadowing other significant patterns related to the nature and order of actions. This sequence-length-driven clustering could reduce the interpretability of our results, as clusters may become mere reflections of session length rather than meaningful groupings based on shared behavioral patterns.

In an attempt to address this, we employed several workarounds such as limiting the sequence length and reducing the granularity of the actions. Although these adjustments allowed us to derive clusters from the data, the clusters obtained were still not particularly interpretable.

The interpretability challenge arose from the difficulty in translating the patterns learned by LSTM, which are embedded in a complex, high-dimensional space, into human-understandable insights. Furthermore, the modification in sequence lengths and action granularity could potentially distort the true user behavior patterns, causing a loss in the quality and authenticity of the clustering results.

As a result, while these experiments with LSTM provided valuable insights into the challenges of applying sequence learning to our data, the outcomes led us to explore alternative methods that could offer a better balance between model performance and result interpretability.

Final Results (spanning 60 days of logs)

SPM

In our Sequential Pattern Mining (SPM) analysis, we began by applying the KMeans clustering algorithm on the UMAP embeddings. By utilizing the 'elbow' method as seen in 10, we identified the optimal number of clusters to be either 4 or 5. We decided to proceed with five clusters. The attributes of these clusters are as illustrated in Table 2

Each cluster shows unique characteristics, with some clusters demonstrating high levels of user activity, shown by shorter time intervals between actions and longer session lengths. In contrast, other clusters feature users who exhibit a more deliberative approach, taking longer periods between actions and accessing a higher number of documents.

Despite these clusters not being our final ones, they provide a valuable glimpse into the diverse behaviors and habits of our user base, setting a useful foundation for further analysis in our Sequential Pattern Mining (SPM) investigation.

We then performed a second round of clustering. In this iteration, we used the clusters obtained from the first round as features, along with several other parameters such as search count, document access count, search to document access ratio, session length, unique actions, average actions before

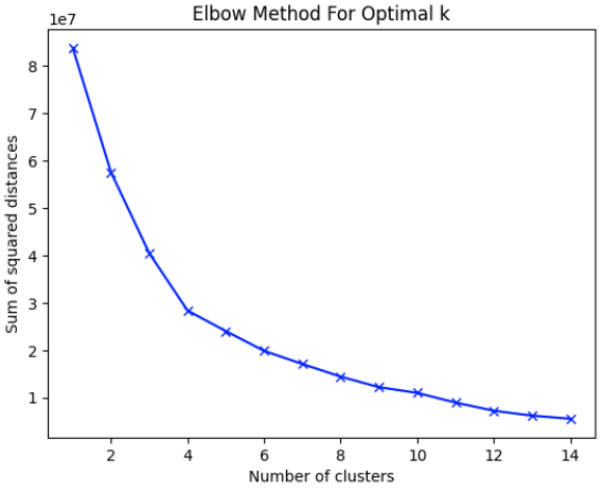

Figure 10

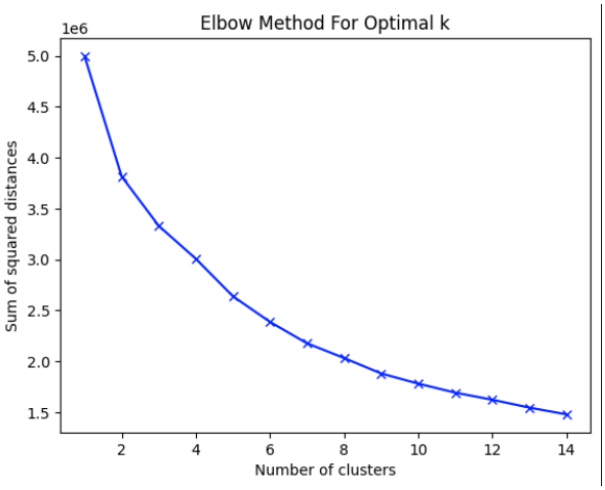

Figure 11

Cluster ID	Size	Mean Session Length	Avg. Time between Actions	Avg. Session Duration	Avg. Document Accesses	Actions before New Document Access	Avg. Searches
0	82708	32.42	39.17	925.25	4.26	7.78	0.07
1	95142	49.43	23.38	940.96	4.79	10.88	29.27
2	143909	31.24	33.52	642.83	3.41	8.26	3.96
3	128739	49.71	24.31	849.99	2.90	13.63	32.82
4	104856	31.56	89.55	1123.81	5.66	5.74	14.05

Table 2: Attributes of the clusters obtained from the first round of clustering

document access, the average time between actions, and session duration. These features were standardized before application.

Following a weighted KMeans approach with added importance on the previous clusters, we identified eight clusters as the optimal number using the elbow method again as seen in 11. . The attributes of these clusters are summarized in Table 3 and can be visualized in the following graphs; Figure 12 which contains the distribution of the actions within each cluster and Figure 13 which can be found in the Appendix and contains multiple histograms describing the features of each cluster.

The next section provides a comprehensive overview of the characteristics and behaviors associated with each identified cluster from the second round of clustering. Each cluster is distinct, exhibiting unique traits and usage patterns that reflect the diverse ways users engage with the platform. The descriptions capture the main attributes of the sessions within each cluster, while the behavior column offers interpretations of the user's activities based on these attributes.

Cluster ID	Size	Mean Session Length	Avg. Time between Actions	Avg. Session Duration	Avg. Document Accesses	Actions before New Document Access	Avg. Searches
0	126965	17.09	35.75	392.24	2.32	3.09	9.96
1	241284	22.43	29.37	498.16	2.01	4.89	2.42
2	13893	104.56	14.14	1376.64	2.56	51.51	41.84
3	7652	128.28	25.67	2770.87	62.95	2.11	9.84
4	11806	12.59	636.02	5263.35	6.04	1.50	1.40
5	38369	131.53	19.63	2453.25	10.71	12.78	82.37
6	13225	79.92	11.49	874.83	1.21	11.86	67.54
7	102160	52.08	23.24	1053.85	4.59	9.11	24.32

Table 3: Attributes of the clusters obtained from the second round of clustering

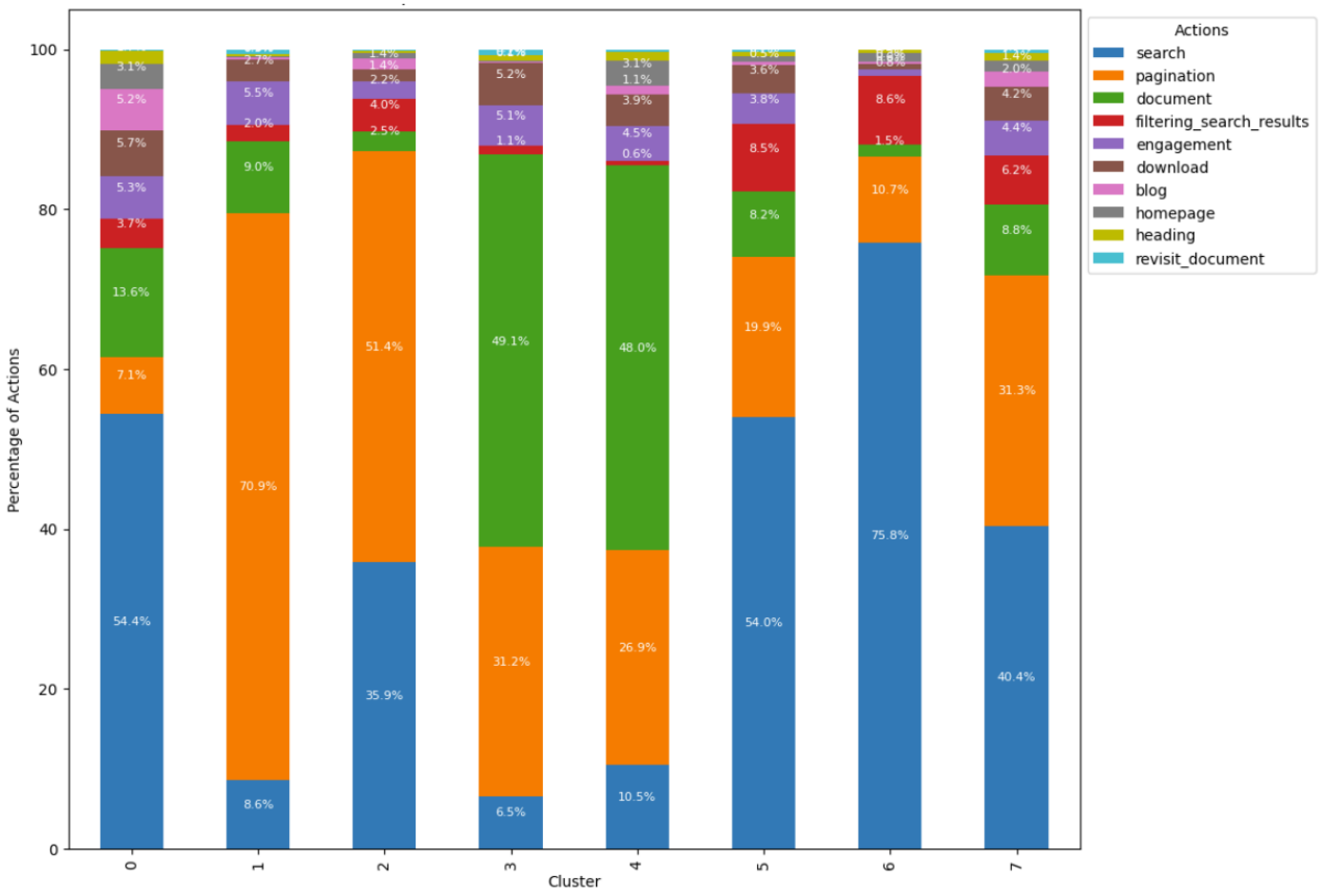

Figure 12: Top 10 action distribution within each cluster

- **Cluster 0:**
  - Description: This cluster is relatively large, accounting for 23% of all sessions. Yet, these sessions are somewhat brief, consisting of an average of 17 actions each. The primary activity within this cluster is searching, followed by document access and engagement activities. Notably, this cluster shows few pagination actions. It has the shortest average session duration across all clusters. The average time between actions is medium-range compared to other clusters. This cluster also stands out for having the highest frequency of downloads, blog navigation, and heading navigation across all clusters.
  - Behavior: The substantial number of searches combined with the frequent blog and heading navigation seems to point towards casual or exploratory users. These users may not have a specific target in mind when interacting with Gallica. They exhibit lower levels of content engagement, implying that they may enjoy simply browsing or exploring the platform's offerings rather than conducting focused research or in-depth reading.
- **Cluster 1:**
  - Description: This is the largest cluster, containing 43% of the sessions. The average session length here is 22.43 actions, which is lower than most other clusters. The most distinguishing characteristic of this cluster is its extremely high pagination percentage (70.9%). Moreover, the average time between actions is average, suggesting a relatively high pace of interaction for pagination action. These users also engage with the document actions, engagement and download actions to a small extent, and have a low percentage of search actions.
  - Behavior: Given the high pagination rate and the small amount of engagement and document actions, these users seem to be engaged in rapid reading or scanning of documents on the platform. The relatively low percentage of search actions may indicate that these users are finding what they need without requiring extensive search activity, possibly revisiting familiar documents.
- **Cluster 2:**
  - Description: Cluster 2 accounts for 2.5% of sessions and is characterized by its long session length, averaging 104.56 actions, the third-longest of all clusters. The average time between actions is the second-shortest, showing high activity. The key actions of these users are search (35.9%) and pagination (51.4%). The very low amount of document access actions is also noteworthy (2.5 documents per session)
  - Behavior: Given the high session length, the low low average time between actions, and the relatively high average session duration, these users are deeply engaged with the platform and possibly doing extensive research or studying a specific topic or document. The high rate of search and pagination actions further suggests an active search for and engagement with content.
- **Cluster 3:**
  - Description: This cluster is very small and represents 1.38% of the sessions. The users in this cluster have the second-longest session length (128.28 actions) and second-highest session duration, indicative of intense and extended usage. They also have the highest average document accesses (62.95 accesses per session) and the lowest search count among all clusters. Document access accounts for 49.1% of their actions, with a significant proportion also dedicated to pagination (31.2%).
  - Behavior: The high average document access, coupled with a high session length and duration, suggests these users are likely deeply exploring and reading the content available

in the documents without knowing preemptively what they are looking for. They might be engaged in in-depth research or exploration, reading multiple documents per session.

- **Cluster 4:**
  - Description: Cluster 4 contains 2.13% of the sessions and stands out for its extremely high average time between actions (636.02 seconds), the highest among all clusters. The average session length is the shortest (12.59 actions). The users of this cluster do a few searches and access a small number of documents without engaging too much with the content.
  - Behavior: Users in this cluster display sporadic activity, reflected in the notably high average time between actions. The relatively short session length implies that these users may be popping in and out of the platform, possibly as they juggle other tasks simultaneously. Despite the lower frequency of searches and document accesses, the documents they do engage with appear to be carefully selected and scrutinized. This suggests that users in this cluster might be using the platform to retrieve specific pieces of information, such as citations or direct quotes, to supplement another ongoing task or project. Their activity on the platform, while infrequent, seems highly targeted and purposeful.
- **Cluster 5:**
  - Description: Cluster 5, encompassing 6.91% of the sessions, is noteworthy for having the highest average session length of all clusters, with approximately 131.53 actions per session. The relatively brief average time between actions suggests a pattern of consistent, active usage. This cluster also records the second-highest average number of searches (82.37) among all clusters and displays a relatively modest frequency of page changes, making up only 19
  - Behavior: The extensive session length combined with a high average number of searches points towards users who are particularly active and engaged with the platform, likely conducting in-depth research or comprehensive document exploration. The dominant role of search actions (54% of all actions) underscores a clear emphasis on locating specific documents or pinpointing particular information. This suggests that the users in this cluster might be undertaking tasks like building a comprehensive corpus of data or conducting meticulous research on specific topics, rather than merely browsing or engaging briefly with the platform.
- **Cluster 6:**
  - Description: Cluster 6 comprises 2.38% of all sessions and is distinguished by its moderately high session length (approximately 79.92 actions on average). A notable feature of this cluster is the notably short average time between actions, the smallest among all clusters, indicating a rapid, highly active interaction pattern. Despite this, the average session duration is relatively brief. Of particular significance in this cluster is the minimum average number of documents accessed, the lowest among all clusters, contrasted with an exceptionally high average number of searches (67.54), the second highest of all clusters.
  - Behavior: The combination of a high volume of search actions and the shortest average time between actions may suggest an aggressive use of the search functionality. It's plausible that these users are either making a rapid attempt to locate specific information or navigating through large amounts of content at high speed. Additionally, this behavior might also hint at automated search processes that have managed to circumvent bot detection measures, perhaps employed to scrape or harvest data from the library at an accelerated pace.
- **Cluster 7:**

- Description: This cluster is the third-largest, making up 18.4% of the sessions. It is characterized by a balance of activities, with a noticeable proportion of search actions (40.4%) and pagination (31.3%). The session length is average (52.08 actions), while the average time between actions and session duration are relatively short compared to other clusters.
- Behavior: This balanced mix of search and pagination suggests these users are actively finding and reading content. They may be engaged in both searching for information and reading or scanning documents, showing a versatile use of the platform.

Upon close examination of the sequences nearest to the centroids, as illustrated in Table **??**, we observe a discrepancy between our expectations for each cluster and the actual sequences. This divergence implies that our clustering algorithm may not be adequately capturing the inherent structures or patterns within our data.

A noteworthy observation is the significant variation in the number of subsequent actions within these sequences. This variability hints at the complexity and diversity of the sequences within each cluster, suggesting that a simple centroid-based clustering might not be sufficient to capture these nuances.

Consider Cluster 1 as an example. The sequences that are closest to the centroid predominantly involve 'downloads' and 'homepages'. However, these actions represent only a small fraction of the overall actions within this cluster. They don't even account for the most frequent actions. This observation is counter-intuitive and suggests that the centroid might not accurately represent the typical behavior within the cluster.

Another interesting example is how the sequence "Document x5 " can be one of the sequences closest to the centroids of Cluster 0 and Cluster 4, even though these clusters aren't that similar.

The disparity indicates potential limitations of our current method, Sequence Pattern Mining (SPM). While SPM has proven effective in capturing frequent patterns within sequences, it seems to struggle with classifying outlier sequences or those with less frequent patterns. This is especially problematic when these outliers become the closest sequences to the centroid, hence misrepresenting the cluster.

7 Critical review: Limitations of proposed methods

Our proposed methods used to identify behaviors, while providing useful insights, are not without their limitations.

To start, a major limitation to our analysis lies in the inherent nature of the data we are dealing with - server-side logs of user requests. It is essential to remember that these logs do not precisely reflect user actions. For instance, when a user opens multiple documents in quick succession, we only capture the requests sent to the server. This data might give us the impression that the user is rapidly accessing and potentially scanning many documents, while in reality, they may be opening all documents at once to read them one by one.

Moreover, when a user switches between different tabs while using the platform, the sequence of actions that we capture in the logs does not accurately represent the user's experience. For example, a user may perform a search, open several documents in new tabs, and then switch between these tabs to read the content. From a log perspective, these document access actions appear clustered together, while the actual user interaction is interspersed with reading time.

These discrepancies between log data and actual user behavior mean that our approach inherently has blind spots and might not fully capture the breadth and depth of user interactions with the platform. Any interpretation of the results must take into account these limitations.

Concerning the use of SPM, firstly, the minimum occurrence value, which is a crucial parameter for SPM, was computationally intensive to test for lower values. Consequently, we had to settle with an assigned value. However, determining the right value is not an intuitive process, mainly because the basic or fundamental pattern of a cluster and how often it repeats itself cannot be preemptively known. There's a trade-off between identifying more detailed patterns with a lower minimum occurrence value and managing computational complexity, which can influence the quality and utility of the resulting clusters.

Secondly, our dataset's sequences were often repetitive. For instance, successive searches or page changes were very common. While repetition is an inherent part of many activities and thus not surprising to find in user behavior data, it can dilute the meaning of individual patterns and make it harder to distinguish truly significant patterns from noise or incidental repetitions. This over-emphasis on certain actions could overshadow more nuanced behaviors and trends that might be of interest. This is especially observed on SGT, where its occasional overreliance on identical sub-patterns can pose challenges. In our opinion, exploring alternative data formatting methods and utilizing the presence of these sub-patterns differently could lead to the discovery of additional interesting behaviors and enhance the quality of clustering results.

Thirdly, judging the effectiveness of our clustering process is challenging since it is inherently a subjective process. The interpretation relies heavily on observing multiple visualizations and aligning those observations with pre-conceived or expected behaviors. This situation might lead us to seek confirmations of our assumptions in the data rather than uncovering truly emergent patterns. In essence, we are attempting to find support for specific behaviors in the data, rather than discovering and defining behavior based on the data alone. This approach might inadvertently introduce biases or limit our understanding of the complexity and variety of user behavior.

Lastly, while patterns can tell us about the order and sequence of actions, they inherently lack temporal context. The only ways we've incorporated a sense of time were through the 'session duration' and 'average time between actions' features. These features give some information about time but do not capture the full temporal complexity of the actions. For instance, two users can have the same sequence of actions, but if one user performs the actions in a much shorter time period, their intention

or experience could be significantly different. Further methods and models might be needed to fully capture and take into account this temporal aspect.

In conclusion, while SPM and clustering have provided valuable insights into user behavior on the platform, these methods come with important caveats and limitations that should be considered when interpreting the results and planning for future research.

8 Future Research

The potential for future research in understanding user behavior on Gallica is extensive.

One key area to investigate is the type of documents users interact with. Gallica boasts a broad array of document formats, such as monographs, booklets, manuscripts, images, and audio files among others. Each of these formats offers a unique engagement experience. For example, users may interact differently with a monograph compared to an audio file. Detailed analysis of these interactions can inform enhancements to user interface design or the refinement of content suggestion algorithms.

Investigating search terms could offer additional valuable insights. By examining the correlation between users' search terms, the search results returned, and the documents users ultimately access, we could gain a clearer understanding of the user journey. Such research could reveal whether users find the information they seek and whether the search functionality adequately meets their needs. These insights could inform improvements in the platform's search algorithms, leading to enhanced user experiences.

Understanding how users engage with suggested documents could also shed light on their navigation habits. Do users generally follow the platform's suggestions, or do they prefer to independently search for documents? This line of inquiry could provide insights into the efficacy of the current recommendation algorithm, and potentially pinpoint areas for improvement.

Adding another layer of depth to these explorations, the referrer of each user request can be included in our analyses. Evaluating where users are navigating from - be it from within Gallica, from external sites, or from direct entries - may illuminate the influence of various pathways on user behavior. For instance, users arriving from outside Gallica may demonstrate specific patterns of engagement that differ from those of users navigating from within the platform. Incorporating this facet into our analysis can enrich our understanding of user engagement and inform further optimizations of user experience.

Furthermore, a temporal analysis of user behavior could reveal more nuanced trends. Analyzing patterns of user activity across different times of the day or days of the week could provide additional insights into user engagement.

Moreover, future research could delve deeper into users' repetitive actions. Identifying and studying common sequences of actions could unveil typical user pathways on the platform. This knowledge could help streamline navigation or identify common challenges in the user journey.

Finally, the exploration of user actions as a Markov chain holds great promise. By examining how a previous action affects the probability of subsequent actions, we could identify potential behavioral changes within the same session. Such an approach could reveal the dynamism of user interactions on the platform and inform the design of adaptive interfaces or features that respond to evolving user behavior.

These avenues of future research all aim at a more comprehensive understanding of user behavior, driving continuous improvement in user experience on Gallica.

9 Conclusion

In conclusion, this report presents the work conducted in filtering and processing user requests to extract meaningful actions and develop user sessions. Our effort focused on creating sequences of actions to better understand user behavior.

Three different methods were explored for the task: LSTM, SGT, and SPM. LSTM was found to be unsuitable due to its high sensitivity to sequence length. SGT performed well on smaller datasets but led to an excessive number of clusters when applied to larger data. SPM emerged as the most effective, scaling well and resulting in eight behaviors that could be related to typical user behavior:

- Casual Browsers: These users primarily engage in search activities and occasionally access documents, spending shorter durations on the platform. Their behavior suggests a casual or exploratory engagement with the platform, browsing content without specific goals.
- Quick Scanners: This group primarily engages in rapid reading or scanning of documents, with a high number of pagination actions and a relatively low number of search actions. It's likely these users are familiar with the platform and revisit known documents.
- Deep Researchers: This small but active group spends a significant amount of time on the platform, searching extensively and engaging with content. They seem to be conducting deep research on specific topics.
- Detailed Explorers: This group spends extensive periods on the platform, often accessing and reading multiple documents in a session. They seem to be exploring the platform content in detail, possibly for comprehensive research purposes.
- Sporadic Users: This group, with the highest average time between actions, often accesses the platform in short, spaced-out sessions. Their behavior suggests targeted use of the platform, possibly to retrieve specific information or supplement other tasks.
- Active Searchers: This group uses the platform intensively, with long sessions characterized by a high number of searches. They appear to be conducting in-depth research or comprehensive document exploration.
- Aggressive Searchers: This group is characterized by a rapid and high volume of search actions with a minimal number of documents accessed. This behavior suggests either a rapid attempt to locate specific information or possible automated processes circumventing bot detection.
- Versatile Users: This group displays a balanced mix of search and pagination activities. They actively find and read content, showcasing a versatile use of the platform.

In short, despite its shortcomings, the study demonstrates a promising approach to classifying user behaviors using sequence mining methods, paving the way for future work in this area.

Acknowledgement

We would like to express our profound gratitude to our supervisor, Simon Dumas Primbault, for his unwavering guidance and steadfast encouragement throughout the entirety of this project. His insightful advice and clarifications were instrumental during moments of confusion, a common occurrence given the project's research-centric nature. His ability to distill complex concepts into comprehensible summaries and diagrams was invaluable to our progress.

Furthermore, we are deeply appreciative of Simon's efforts to broaden our horizons beyond data science. He introduced us to the fascinating field of sociology, illuminating its practices and methodologies, thereby enriching our understanding and experience and making our work significantly enhanced due to this interdisciplinary approach.

10 References

1. Adrien Nouvellet, Valérie Beaudouin, Florence d'Alché-Buc, Christophe Prieur, François Roueff. Analyse des traces d'usage de Gallica : Une étude à partir des logs de connexions au site Gallica. [Rapportde recherche] Télécom ParisTech; Bibliothèque nationale de France. 2017. ffhal-01709264ff

2. Trabelsi, Marwa. "Modélisation des processus utilisateurs à partir des traces d'exécution, application aux systèmes d'information faiblement structurés." Doctoral thesis, La Rochelle Université, École Doctorale Euclide, Laboratoire L3i, soutenue le 1er septembre 2022. Discipline: Informatique et Applications.

3. Gan, Wensheng, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. "A Survey of Parallel Sequential Pattern Mining." In Proceedings of the International Conference on Advanced Data Mining and Applications, pp. 3-18. Springer, Cham, 2021.

4. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H. (Intelligent Database Systems Research Lab, School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada). "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth." Q. Chen, U. Dayal, M.-C. Hsu (Hewlett-Packard Labs, Palo Alto, California, USA). Technical Report.

5. Deveaud, Romain. "Modalités d'accès au savoir ouvert sur les plateformes d'OpenEdition." Aix-Marseille Université, OpenEdition, LIS, romain.deveaud@openedition.org.

11 Appendix

Directory Structure

```
iiif
    ark:
ark:
assets
    static
        javascripts
        images
        stylesheets
services
    ajax
        extract
        pagination
        mode
        action
            download
            search
            infosdetails
            tdm
            share
            caption
    engine
        search
    image
        highlighter
    search
        resume
html
    sites
    modules
    misc
    und
mbImage
    logos
m
    images
    ark:
    js
        lib
        commons
        search
        viewer
    css
blog
    sites
        all
        default
essentiels
    sites
        all
```

Category	Subcategory	Description
Homepage	-	The homepage of Gallica is considered a significant entry point for the user's journey. If the endpoint is an empty string, it means the user is on the homepage.
Download	-	Any request containing 'download' in the endpoint and 'ark' in endpoint 1, or any endpoint that contains 'services/ajax/action/download/' is considered a download request.
-	Document Download	If a download request contains 'services/ajax/action/download' in the endpoint, it is marked as a document download.
-	Page Download	If a download request ends with 'download=1' in the endpoint, it indicates a page download.
Document	-	Any request where 'ark' is present in endpoint 1 and is not a download request, is considered a document request.
IIIF	-	International Image Interoperability Framework (IIIF) requests are denoted by 'iiif' in the endpoint 1. IIIF provides a standardized method for describing and delivering images over the web.
Static HTTP	-	If 'assets' or 'html' are present in endpoint 1 and endpoint 2 is not 'und', the request is deemed a static HTTP request.
Blog	-	Any endpoint 1 that contains 'blog' is considered a blog request.
Services	-	If 'services' is present in endpoint 1 and endpoint 3 does not include 'search', 'pagination', 'action', or 'mode', then the request is classified as a service request.
Search	Simple Search	We classify any endpoint radical containing 'services/engine/search' as a search request.
	Advanced Search	If the endpoint contains 'advancedSearch', it is considered an advanced search.
	Filtering Search Results	If the endpoint contains 'subsearch' or 'restrictedSearch', we consider this as filtering search results.
Pagination	-	Any endpoint containing 'services/ajax/pagination' is considered a pagination request.
Mode	-	If an endpoint contains 'services/ajax/mode' but does not contain 'zoom', it is marked as a mode request.
Zoom	-	If an endpoint contains both 'services/ajax/mode' and 'zoom', it is classified as a zoom request.
Heading	-	If endpoint 1 is 'html' and endpoint 2 is 'und', the request is considered a heading request.

Table 4: Categories of User Preprocessed requests

Parent Action	Action	Description
homepage	homepage	Accessing the homepage
blog	blog_navigation	Navigating within the blog
heading	heading_navigation	Navigating drop menus
Search	simple_search	using the search engine
advanced_search	advanced_search	Using the advanced search engine
filtering_search_results	filtering_search_results	Applying filters to refine search results (following simple search)
document	document_access	Accessing a specific document
revisit_document	revisit_document	Revisiting a previously accessed document within the session
pagination	first_page	First accessed page of a document
	prev_page	Pressing the previous page button in a visited document
	next_page	Pressing the next page button in a visited document
	chosen_page	Choosing a page to jump to inside a visited document
engagement	zoom	Zooming in or out of a page
	to_single_page_mode	Switching to single page mode
	to_double_page_mode	Switching to double page mode
	to_vertical_page_mode	Switching to vertical page mode
	to_audio_page_mode	Switching to audio page mode
	to_multi_page_mode	Switching to multi-page mode
download	page_download	Downloading a specific page of a document
	document_download	Downloading a document

Table 5: Further categorization of actions following sessionization

Cluster ID	The Three Closest Sequences to the Centroid
0	- document x2 → heading x3 - blog x3 → document x2 - document x5
1	- download x5 - homepage x5 - pagination x5
2	- document x2 → search x32 → document x1 - document x2 → pagination x40 → search x6 → document x1 - document x2 → pagination x41 → document x1 → pagination x5
3	- document x47 - document x12 → revisit_document x1 → document x43 - document x7 → search x1 → document x35 → search x1 → document x1 → search x1 → document x2
4	- document x5
5	- homepage x2 → search x88 - document x3 → search x28 - homepage x2 → search x20 → document x2 → pagination x13 → search x43 → revisit_document x1 → document x1 → download x1 → document x1 → download x2
6	- search x40 → filtering_search_results x1 → document x1 - homepage x2 → search x14 → filtering_search_results x1 → search x20 → filtering_search_results x1 → search x1 → document x1 → engagement x1 - document x2 → search x36
7	- document x2 → pagination x1 → revisit_document x1 → search x1 → filtering_search_results x1 → search x1 → document x1 - document x2 → pagination x3 → search x1 → heading x2 → revisit_document x1 - homepage x2 → search x4 → document x1 → filtering_search_results x1 → engagement x1

Table 6: The three closest sequences to the centroid for each cluster.

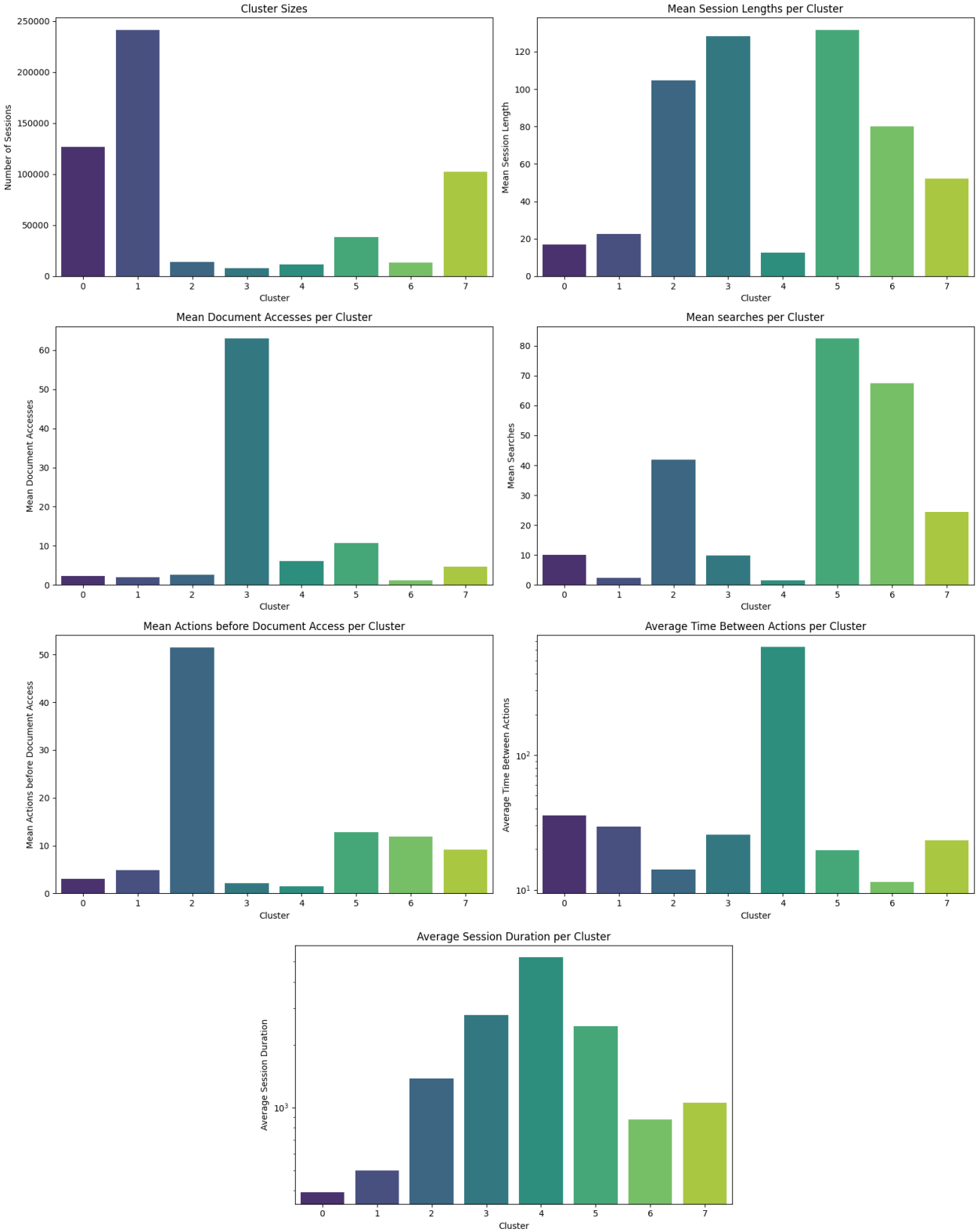

Figure 13: Feature distribution for each cluster

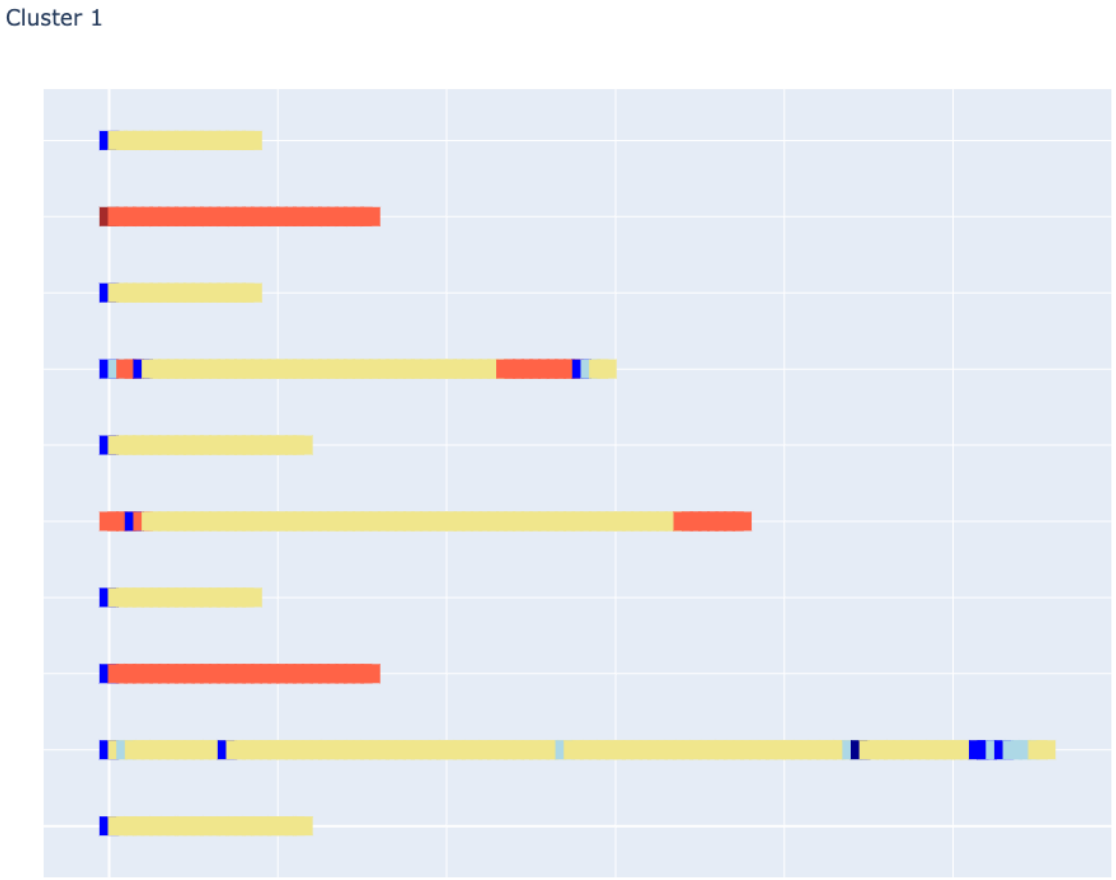

Figure 14: Closest 10 Sessions to centroids for cluster 1 (2 days)

Figure 15: Closest 10 Sessions to centroids for cluster 2 (2 days)

Figure 16: Closest 10 Sessions to centroids for cluster 3 (2 days)

Figure 17: Closest 10 Sessions to centroids for cluster 4 (2 days)

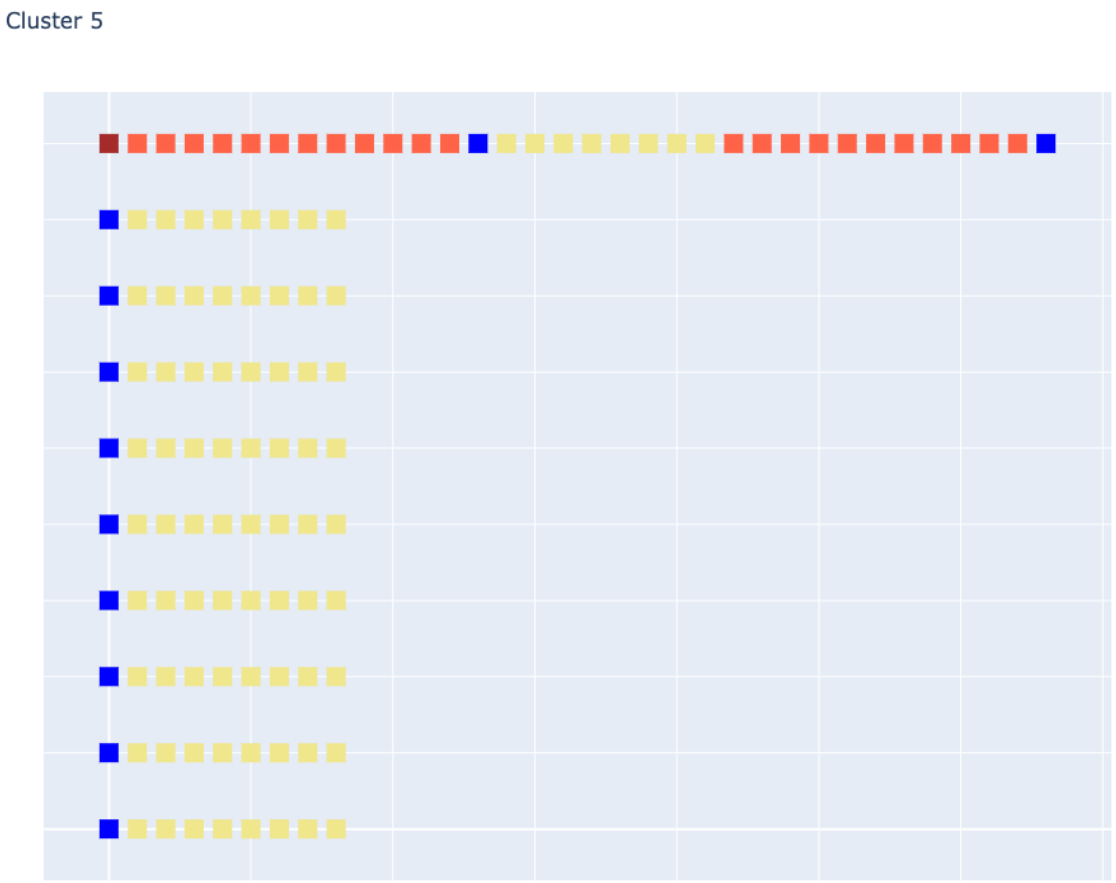

Figure 18: Closest 10 Sessions to centroids for cluster 5 (2 days)

Figure 19: Closest 10 Sessions to centroids for cluster 6 (2 days)

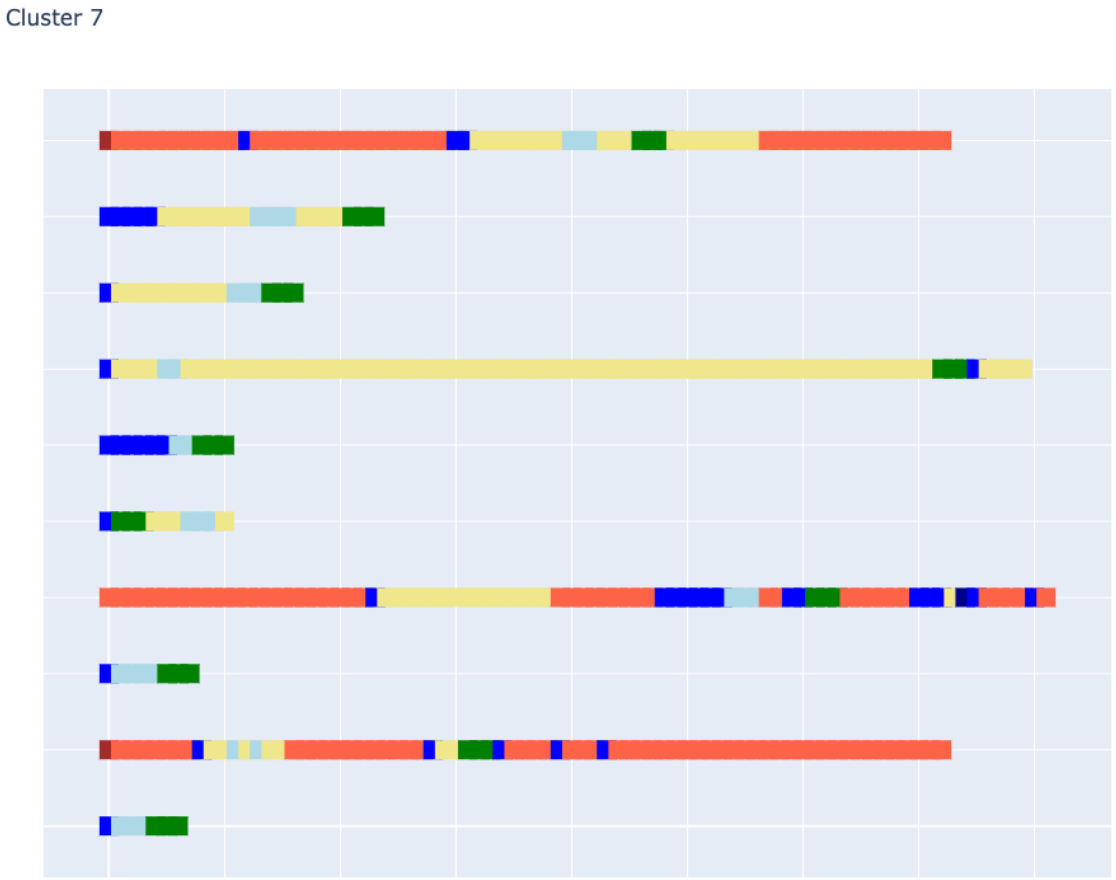

Figure 20: Closest 10 Sessions to centroids for cluster 7 (2 days)

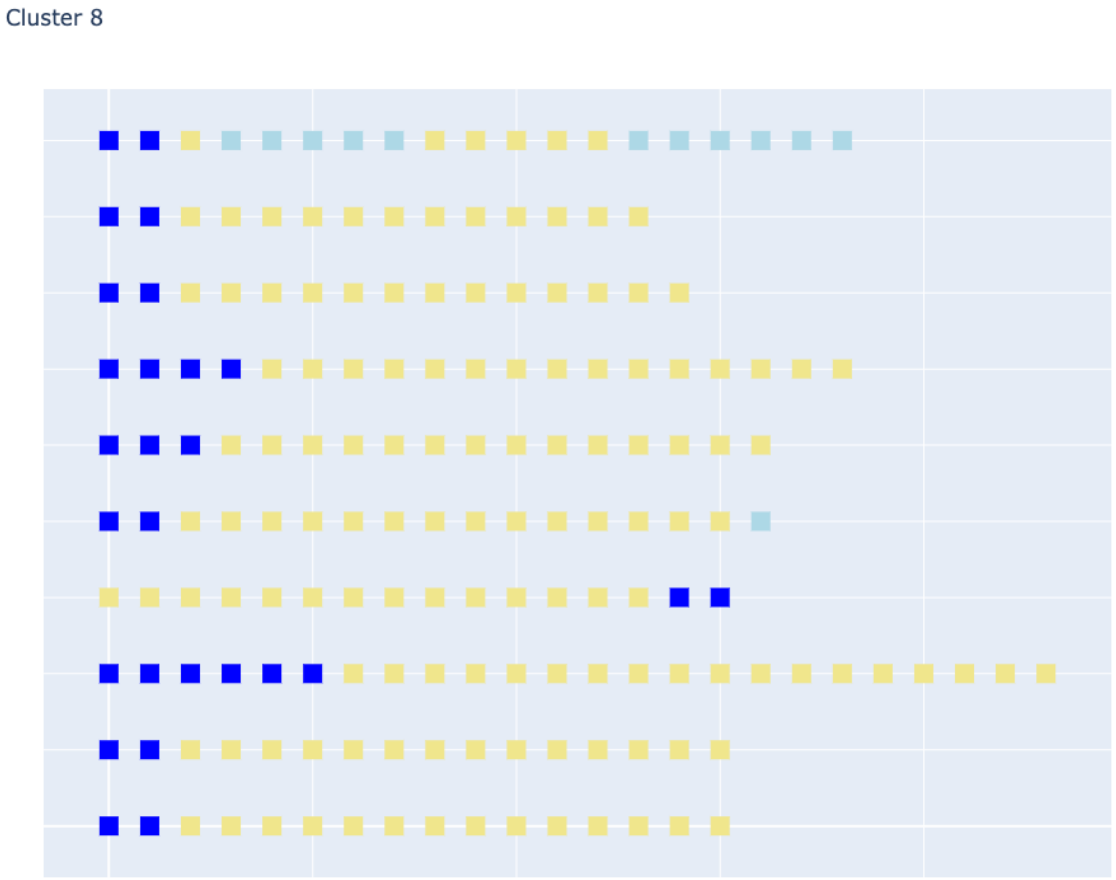

Figure 21: Closest 10 Sessions to centroids for cluster 8 (2 days)

Figure 22: Closest 10 Sessions to centroids for cluster 9 (2 days)

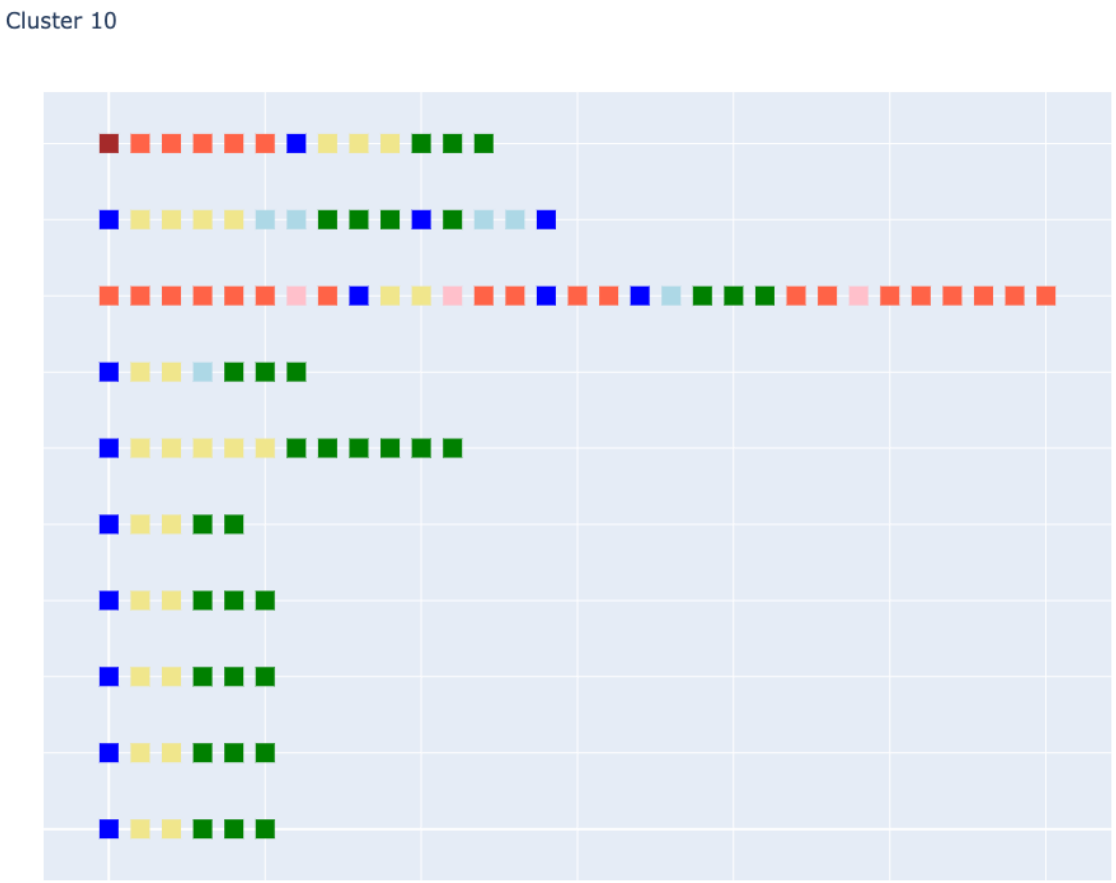

Figure 23: Closest 10 Sessions to centroids for cluster 10 (2 days)