Theorizing Routines with Computational Sequence Analysis: A Critical Realism Framework

Abstract

We develop a methodological framework to develop process theories on routines by leveraging large volumes of digital trace data following critical realism principles. Our framework starts with collecting and pre-processing digital trace data, corresponding to the empirically observed experience of critical realism. At the second and third steps of the framework, we identify a finite set of similar repetitive patterns (routines) through computational analysis. We accomplish this by combining frequent sub-sequence mining and clustering analysis to transform empirical observation into a set of routines that correspond to actual happening in critical realism. Then we employ a retroduction approach to identify generative mechanisms of the routines. At the final step, we validate the generative mechanisms by evaluating proposed processual explanations and/or eliminating alternatives. We provide an illustrative example of developing a process theory with regards to the collaboration pattern in Wikipedia.

Keywords: *Process studies, routines, sequence analytics, critical realism.*

INTRODUCTION

Process studies offer deeper theoretical insights into how events and activities unfold over time. They allow scholars to uncover the underlying generative mechanisms that produce events and activities that are directly observable (Langley 1999; Pentland 1999; Van De Ven and Poole 1995). A stream of process studies focuses on the organizational routines. Routines are regular features of the sequences of activities and events which form repetitive patterns in organizing (Gaskin et al. 2014; Salvato and Rerup 2011; Pentland 2003). While routines allow scholars to understand the dynamic aspects of how actions taken by actors shape and are shaped by unobservable social structures, we lack methodologies to develop processual theories centered on the generative mechanisms of routines leveraging large-scale digital trace data. Our paper provides a computational methodological framework to analyze digital trace data grounded with critical realism to discover routines and identify underlying generative mechanisms.

As digital technologies are used extensively in organizing, scholars who study routines can leverage digital trace data. Digital trace data will be particularly useful in studying routines in new forms of organizing, such as online communities. However, theorizing routines with digital trace data poses two challenges: discovering routines and identifying generative mechanisms from large volumes of digital trace data.

Firstly, identifying recurring patterns of sequenced organizing activities and/or events becomes much more challenging with a large volume of digital trace data. Feldman and Pentland (2003: 95) characterized routines as “repetitive, recognizable patterns” (Pentland and Rueter 1994). Traditionally, scholars “construct [process] theory from qualitative “process data” collected in close contact with real contexts” (Langley 1999: 705). In such a setting, routines are usually

shorter and more standardized with a clear beginning event and ending event (Pentland and Rueter 1994, Pentland et al. 2010), and relatively limited variations. On the contrary, digital trace data has fine granularity with a volume that exceeds any human can process effectively. For example, Github.com records more than 20 different event types created by the developers 24/7, generating more than 200,000 events each day. As online communities are "fluid" (Faraj et al. 2011), the interactions are much more diverse in actor, activity, and time. As such, how to identify "repetitive and recognizable" patterns becomes an extremely challenging task for process scholars.

Secondly, such an increased amount of data poses new challenges to the analytical capabilities and our fundamental understanding of the nature of the data and the way to analyze it. In the literature, as Pentland et al. (2010) argued, routines have been assumed to be directly observable and can reflect reality (Gaskin et al. 2014; Goh and Pentland 2019; Hansson et al. 2018; Lazer et al. 2009; Lindberg et al. 2016; Pentland and Hærem 2015; Schecter et al. 2018). These studies tend to focus on measuring the variance between different sequences to investigate the theoretical causes of the variance. However, as digital trace data is often collected from ongoing, sometimes never-ending, processes at different temporal stages, the results from comparing the entire sequence as the unit of analysis can be inappropriate and often impossible.

We propose a critical realism-based methodological framework for conducting *sequence analytics* that leverages large-volume sequential digital trace data to advance processual theories by identifying routines and the underlying generative mechanisms. By sequence analytics, we refer to a systematic way of computationally analyzing digital trace data as a sequence of events to identify hidden temporal structures of the actual happening to uncover the underlying generative mechanisms that produce those events. Sequence analytics allows scholars to explore an

unfolding sequence of events to identify routines and understand the process dynamics (Abbott 1990; Gaskin et al. 2014). More specifically, we propose using *sequential pattern mining* and *clustering* with a critical realist perspective to identify the underlying generative mechanisms that give birth to directly observable instances of routines. By relying on critical realism, we use computational tools as a device (i) to observe "empirical" reality by explicating events and (ii) to identify demi-regularity (routines).

Our sequence analytics approach contributes to the growing body of computational studies on information systems (Gaskin et al. 2014; Lazer et al. 2009; Lindberg et al. 2016; Pentland and Hærem 2015; Schecter et al. 2018) by providing concrete computational tools and a philosophical foundation of discovering routines and the underlying generative mechanisms.

After the introduction and background, we put forward an integrated methodological framework for computational process study. We explain the framework step-by-step with an empirical illustration. We conclude the paper by discussing the various threats to validity and reliability and the limitations.

LITERATURE REVIEW

Routines in Organizing and Computational Process Studies

Extant process studies are predominantly conducted with inductive, qualitative approaches based on data often collected through in-depth interviews, direct observations, and longitudinal ethnographic studies (Langley 1999; Langley et al. 2013; Langley and Tsoukas 2016). However, a group of scholars focusing on routine often supplement qualitative interpretations with complementary quantitative or even computational analysis. Frequently, those studies treat routines as the unit of analysis (Pentland and Feldman 2005), and computational analysis is

usually applied to quantify the routines' characteristics. Most of those studies (Gaskin et al. 2014; Goh and Pentland 2019; Hansson et al. 2018; Pentland and Hærem 2015; Schecter et al. 2018) focus on the traditional organizing setting, where routines can often be easily identified as completed processes of some standard or institutionalized practices. In those studies, data is usually collected through observations or interviews.

As digital trace data has increasingly become an important source of observing changes in society, Information Systems (IS) researchers have also started using computational methods to analyze digital trace data. The computational approach refers to efforts to develop a deeper understanding of individual and group behaviors using large scale digital trace data produced by the computing devices and systems used by human actors (Chang et al. 2014; Giles 2012; Lazer et al. 2009). While most computational studies focus on large-scale network analysis (Giles 2012; Liben-Nowell and Kleinberg 2007; Onnela et al. 2007), we have started to see computational approaches analyzing sequential data. We can group the contemporary approaches to computational analytics in process studies roughly into two streams.

Researchers in the first stream use computational analytics to generate key measurements from process data to operationalize some constructs in their variance models. This approach is seen as the most straightforward way of applying sequence analysis in process studies (Langley and Tsoukas 2017). Gaskin et al. (2014) propose a computational approach to model process data as sequences and calculate various measurements through sequence analysis. Lindberg et al. (2016) measure routine variations using sequence analysis to conduct statistical tests to examine the relationships between these variations and interdependencies in online communities. Li and Zhou (2017) apply sequence analysis to measure the variability of the open-source software development process and then explored the relationship between process variability and developer mobility through econometric analysis. This line of studies is more like an updated

version of a traditional quantification strategy (Garud and Van de Ven 1992; Langley 1999; Van de Ven and Polley 1995) where processes are quantified and abstracted into numeric constructs, which essentially transform a process study into a variance study.

Another stream of researchers mostly follows an inductive approach and generally use computational analysis to complement traditional interview-based approaches. For example, Berente et al. (2019) discuss how to leverage big data for grounded theory development by incorporating computational analysis. In this stream, scholars use computational analysis mostly to automate or assist discovering the concepts (Gioia et al. 2012). Then they develop process models by inductively formulating the dynamics among the discovered concepts. For example, Vaast et al. (2017) discover three major user roles in using social media (Tweeter) by first performing a clustering analysis, then modeling and identifying typical interaction patterns among different user roles using a network motif analysis. Based on the result, they then proposed a new explanatory theory.

It is also worth noting that many computational process studies also use network tools, instead of sequence analysis tools, to explore the dynamics within the process. For example, Vaast et al. (2017) use interaction networks among the different types of actors involved to identify motifs that represent typical interactions. Goh and Pentland (2019) similarly create activity networks using the sequence of activities collected from EMR logs. Meanwhile, Schecter et al. (2018) develop a relational event network method to analyze the sequence of relational events among actors, leveraging social network analysis. However, the richness of digital trace data is generally lost during the construction of these different types of networks, as complex events are often represented by nodes of a single attribute (e.g., actor (Vaast et al. 2017), action (Goh and Pentland 2019), or relational events (Pilny et al. 2016, Schecter et al. 2018)). As a result, the

interpretation of the analysis result is usually relatively simple and is limited to the remaining attributes.

Scholars are recently increasingly interested in applying process mining techniques (van der Aalst et al. 2007) to study organizing routines (Grisold et al. 2020). Process mining is a family of computational algorithms that use digital trace data (e.g., event logs) 1) to build process models (process discovery), 2) to check if process instances follow the prescriptive process model (conformance checking), and 3) to identify bottlenecks that create a negative performance of processes (van der Aalst et al. 2011). Given its origin in the workflow management system discipline, the process mining tool is designed to support workflow managers in developing workflow schema, monitoring the workflow instances according to the schema, and handling exceptions in the middle of the execution of workflow instances (Andrews et al. 2018). Noting the conceptual similarity between workflow schema and routines, Grisold et al. (2020) discuss how process mining techniques can be used to develop theories about endogenous-evolutionary and exogeneous-punctuated changes in routines. However, process mining techniques target structured business processes in which the beginning and the end of process instances are easily identified. On the other hand, many organizational processes, particularly in virtual settings, are unstructured, and the beginning and end of process instances can be obscure. The literature is still at the early stage (see Grisold et al. (2020)). There is a lack of any attempt to develop theories considering the multi-layered nature of routines.

Sequence Analytics

Sequence analytics is a branch of computational tools developed for conducting sequence analysis. Social science scholars have used computational sequence analysis for over three

decades (Abbott 1990; Abbott and Forrest 1986). The most commonly used sequence analysis uses multiple sequence alignments, sometimes also called the *optimal matching method* (Abbott and Tsay 2000). By aligning and comparing different process sequences, scholars can categorize the sequences based on their similarity and identify patterns based on that categorization. These studies generally examine the whole course of the change process, as scholars are typically interested in the sequence of events in the context of actual history. Examples of patterns identified through this approach include career progress patterns (Abbott and Hrycak 1990; Blair-Loy 1999; Halpin 2010; Halpin and Cban 1998), life trajectory patterns (Stovel and Bolan 2004; Wilson 2001), vacation patterns (Bargeman et al. 2002; Shoval et al. 2015), management's cognitive schema changes (Bingham and Kahl 2013), organizational change rhythms (Klarner and Raisch 2013), discursive history (Maguire and Hardy 2013), design processes (Gaskin et al. 2014), and open-source development processes (Lindberg et al. 2016).

Sequential pattern mining is a less frequently used sequence analysis approach that focuses on identifying the latent segment, termed a frequent sub-sequence, instead of comparing whole sequences. With sequential pattern mining, scholars seek to discover frequent sub-sequences as patterns in sequence data, which present the sequences of ordered events using a concrete notion of time. Events in time, codons, or nucleotides in amino acids, website traversal, computer networks, and characters in a text string are examples of where the existence of sequences may be significant and where the detection of a frequent sub-sequence might be useful (Mooney and Roddick 2013). Such frequent sub-sequences offer opportunities to identify "repetitive, recognizable patterns" (Feldman and Pentland 2003, p. 95).

Computer scientists first developed sequential pattern mining to discover latent sequential patterns (Agrawal and Srikant 1995). Natural scientists use it to discover sequence motifs[1] from biological sequence analysis (D'haeseleer 2006a). Although it was discussed in the social science literature shortly after it was invented (Abbott and Barman 1997), this approach has never been widely adopted by other scholars (Cornwell 2015: p.195).

In this study, we propose applying sequential pattern mining to better understand routines in organizing leveraging digital trace data. In so doing, we draw on critical realism as the foundation of the computational approach. We argue that we must understand the meaning of directly collected trace data in a broader social context.

Critical Realism

Critical realism is an alternative philosophical paradigm to both positivism and interpretivism in conducting social science research (Bhaskar 2013; Wynn and Williams 2012). It focuses on the causality of social and natural phenomena and tries to explain how and why such phenomena occur by identifying generative mechanisms.

The ontological stance of critical realism is based on independent reality and stratified ontology. Independent reality indicates that the reality of our world exists independently of our perceived knowledge. With independent reality, scientific research is concerned with understanding things and their causality in two dimensions: intransitive and transitive (Bhaskar 1998). Things and entities in the intransitive dimension are real, while those in the transitive dimension are perceived

[1] Sequence motif in biology is the short but recurring pattern in DNA sequence that may carry special biological functions (D'haeseleer 2006b).

and are therefore based on human perception. Our knowledge is created in the transitive dimension and based on observed events, while our theories about those events are caused by intransitive entities in the social and natural worlds. Therefore, knowledge in the intransitive dimension is not absolute and can be revised as more experience of events is accumulated via human observations or computing tools on intransitive entities in the world.

The two-dimensional view of the world leads to a stratified ontology with three domains: *real*, *actual,* and *empirical* (Bhaskar 2013). The *real* domain includes entities, structures, and the causal powers inherent in them. Events are created when the causal powers of entities or structures are enacted. The events created by causal powers may be different when the causal powers interact with the contextual factors of the settings. The domain of the *actual* is a subset of the domain of the real and includes events created by causal powers. These are events that have actually occurred, whether they are observed by humans or not. The domain of the *empirical* is a subset of the domain of the actual and includes events that are experienced by human observers. Empirical data often differs from the actual data because of the observer's interference (Danermark et al. 2001).

Such an ontological view offers an alternative perspective to the commonly adopted positivist paradigm when analyzing digital trace data generated by human activities. Digital trace data represents "a slice" (Jungherr and Theocharis 2017, p. 99) of the activities that are monitored by information infrastructure and produced as the "by-product" of monitored activities (Howison et al. 2011, p. 769). Digital trace data, therefore, is different from the sample data that is purposefully collected through deliberately designed instruments, which can be seen as having direct access to the real world in positivists' view. By viewing digital trace data as *empirical observations of*

actual events[2] *generated by the underlying real mechanisms* "in a specific contextual environment at a specified time" (Wynn & Williams 2012, p. 793), critical realism recognizes the ontological independence of the empirical observations. It holds an "open systems perspective" that the variations in empirical observations are caused by the underlying mechanisms and structures and the changing contextual conditions (Wynn & Williams 2012). This perspective is especially helpful when scholars need to analyze vast amounts of digital trace data with high variety generated in virtual environments, such as online communities. It requests scholars to go beyond the observed variances and look into the real structures and mechanisms underneath.

In addition to providing a more appropriate ontological view of digital trace data, the epistemological assumptions of critical realism allow IS researchers to develop "context-specific causal explanations of socio-technical phenomena by explicating the specific mechanisms which generate them" (Wynn and Williams 2012, p. 795). Critical realism does not seek to deductively establish causal relationships among measured constructs like positivists do, nor inductively formulate dynamic relationships among identified concepts like interpretivism does (Gioia et al. 2013). Instead, critical realism relies on retroductive reasoning to propose and find the best plausible *explanations* that can make sense of discovered patterns. As such, an iterative and multimethod-based approach to building the mechanisms that generate experiences (observable events) is the key artifact of the critical realism approach. The mechanisms need to be verified based on theory and the empirical data collected.

Pentland and Rueter (1994) propose the influential grammatical model of routines, where routines can be viewed as a set of possible performances for a specific task that is "in part" defined by the

[2] Here, an event refers to a specific happening or action caused by the enactment of underlying mechanisms (Wynn and Williams 2012).

grammar. What is left out of their model is the syntax that governs the construction of a routine. In linguistic theory, syntax is seen as the generative rules that describe the formation of grammatical sentences (Everaert et al. 2015). Feldman and Pentland (2003) propose ostensive and performative aspects of routines. Similarly, Becker (2005) argues that routines should be seen as a multi-level concept following a critical realist's ontological view of three-level stratified reality (despite not directly using the term "critical realism"), suggesting to start from "recurrent action patterns" on the level of "empirical." Therefore, critical realism offers an ideal philosophical view in studying routines, where performances of routines are captured as empirical observations, and routines are actual events generated by underlying mechanisms.

Traditionally, scholars who follow critical realism have primarily used qualitative methods that are not suitable for analyzing vast amounts of digital trace data. Wynn and Williams (2012) propose five methodological principles for case studies derived from critical realism, including explication of events, explication of structure and context, retroduction, empirical corroboration, and triangulation and multimethod. Other scholars have also proposed methodological approaches to critical realism research. For example, Fletcher (2017) proposes a flexible coding and data analysis process according to the critical realism approach. He uses a linear process of identification of demi-regularities, abduction, and retroduction to identify two generative mechanisms (namely, gender ideology at the household and farm level, and corporatization in Canadian culture) that shape the lives of prairie farm women in Canada using two types of data: extensive data (statistical) and intensive data (interview and focus groups in the context of Saskatchewan farm women's experiences). Miller (2015) discusses how agent-based modeling can be used in critical realism to study various topics in organizing. Zachariadis et al. (2013) propose a multi-phase framework for applying critical realism in mixed-methods research.

Henfridsson and Bygstad (2013) use a configurational approach to identify the generative mechanisms of the digital infrastructure evolution.

Despite those early efforts, these methodologies are mainly dependent on manual work to collect and analyze data in different domains (empirical, actual, and real), thus not suitable for a large volume of digital trace data accumulated in real-time. The availability of large volumes of digital trace data together with a new breed of software tools specifically designed to deal with sequence data broadens the scope of the events that would not have been observed by human scientists without such tools and trace data. A new methodological framework based on the computational tools to help scholars to process observed data and then discover the generative mechanism is urgently needed in a digitalized society where the interactions among people are becoming more real-time and complex.

SEQUENCE ANALYTICS FRAMEWORK

An Overview

Broadly following the methodological principles of critical realism set out by Wynn and Williams (2012), we propose a five-step framework for conducting studies of routines using sequence analytics (Figure 1). In particular, we take a process-oriented approach. Howard-Grenville et al. (2016) argue that traditional routine studies take an entitative-oriented approach that considers routines as being similar to computer programming or genes and views them as building blocks that generate stability, efficiency, and a unique capacity within organizations assuming routines are designed to produce invariant outcomes regardless of who performs the routines.

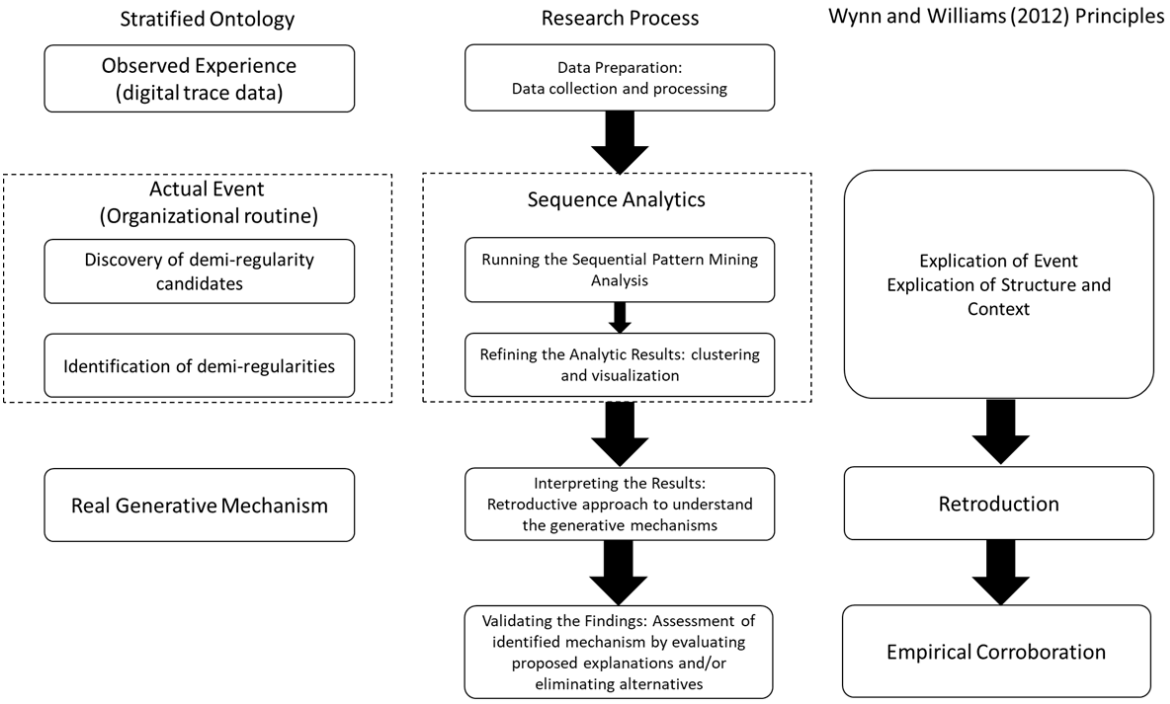

Figure 1 A sequence analytics framework for computational process research

In contrast, the process-oriented approach (Feldman and Pentland 2003) emphasizes human agency involved in routines, which emerge, persist, and change over time depending on the actions taken by human actors. The enactments of routines may differ depending on the contexts. The process-oriented approach to routines assumes a multi-layered structure of routines. The generative layer is different from the surface layer in which the enactments of routines are affected by different structures, actors, and temporal features. Therefore, the process-oriented approach is in line with the principle of critical realism. The variability of routines in the underlying layer can explain the variances of actions observed in the empirical layer.

Our framework starts with collecting and pre-processing digital trace data, which corresponds to the empirically observed experience of critical realism. The second and third steps of the framework are concerned with identifying primitive patterns (i.e., groups of similar repetitive patterns) through computational analysis. From the perspective of process studies, these primitive

patterns serve as a form of routines. Our view is that empirical observations collected from digital trace data are the performances of routines in the domain of the actual, which may vary in different contexts. Observed experiences contain actions taken by actors in certain orders. These are still difficult to translate into routines due to the vast number of observations and the high variety caused by specific contexts. We accomplished this by combining frequent sub-sequence mining and clustering analysis to transform empirical observations into actual happenings. The former analysis identifies latent patterns, while the latter identifies primitive patterns. A latent pattern is defined as a section of sequences that repeatedly appears in different processes; a primitive pattern is a meta pattern that represents a group of similar latent patterns.

Once routines in the actual domain are identified, a retroduction approach is employed to identify the generative mechanisms of the routines. The final step is to validate the findings via assessing identified mechanisms by evaluating proposed explanations and/or eliminating alternatives. Table 1 shows how the sequence analytics method transforms empirical observations into routines from one domain to the next domain compared with the traditional critical realism approach. We argue that triangulation between the traditional and emerging computational methods is an important element that defines successful computational process research.

Table 1 Stratified ontology in traditional versus sequence analytics-based critical realism studies

Stratified ontology	Traditional approach (Wynn and Williams)	Sequence analytics
Observed experiences: observed instances of many possible performances of routines.	Researchers' own observation: e.g., interview-based case study. Low volume, low variety, and less structured	Digital trace data automatically collected by information infrastructure. High volume, high variety, and more structured.
Actual Events: routines formed by subroutines and governed by the underlying mechanisms.	Demi-regularities are identified through manual explication.	Demi-regularities are identified as primitive patterns through sequence mining and clustering.

Real Mechanism: the rules and structures that can generate the routine.	Iteratively retroduce explanatory mechanisms.	Iteratively retroduce explanatory mechanisms.

The Overview of an Illustrative Example

Below, we explain each step of our framework in detail. Along with each step, we demonstrate how the proposed framework can be applied in an empirical study by detailing the operational steps of sequence analytics. We also discuss key considerations for researchers to bear in mind. For the illustrative example, we use data collected from Wikipedia, the largest online collaborative project, with timestamped event logs, focusing on the collaboration activities among different participants. In such a virtual community, articles are being continuously edited by different participants; this makes it impractical to identify routines using traditional approaches and provides an ideal setting to demonstrate the use of our framework. In Wikipedia, there are two types of articles, namely, featured and non-featured articles. A featured article in Wikipedia is one of the "best articles Wikipedia has to offer."[3] Such articles have to meet some very strict criteria that emphasize accuracy, neutrality, completeness, and style.

In this illustrative example, we show how we followed our framework to discover *if and how featured articles are created through different collaboration patterns over time, compared to non-featured articles*. Collaboration patterns in Wikipedia represent repetitive and recognizable sequences of members' editing activities to develop articles and are the core routines of Wikipedia. By comparing the patterns of collaboration over time between featured and non-featured articles on Wikipedia, we use our sequence analytics framework to identify the generative mechanisms that cause certain types of collaborative patterns that result in high-quality Wikipedia articles.

[3] https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

Step 1: Data Preparation

The first step of sequence analytics is data preparation, which includes data collection and data processing. Researchers need to determine the source of the relevant data and secure access to it. Digital trace data is usually downloaded from the source, often by using customized scripts or APIs. Such data provides a means to capture the *empirical* reality. However, not all the data available is relevant to the focal study. Thus, one of the key tasks is to identify the necessary features of the events. This process creates the foundation for researchers to explicate the events further in the next step (Wynn and Williams 2012). Therefore, it is important to select and collect data that can carefully describe the essential features of target events. The raw data collected is often in the format of system logs stored in a database. It may be structured (like HTTP logs) or unstructured (like the interview transcripts, historical archival data, or observational data often used in conventional process studies). The raw data needs to be further processed into a proper format with the desired information for analysis.

The data sources amenable to sequential analytics can include unstructured data like social media posts and articles on online communities like Reddit and structured data such as event logs from workflow management systems. These data may provide the timestamp, actor, and contents of events, which can be transformed into sequences of events. The collected data needs to be evaluated from three aspects as defined in Table 2 to avoid potential biases described in the table.

Table 2 Evaluating data for sequential analytics

Completeness of Event	Is the system designed to capture all kinds of events or only specific types of events?	It may cause "selection bias" if the system captures only certain types of events or different events are captured disproportionately. For example, the system

		always captures event A but only selectively captures event B.
Temporal Accuracy	Is the system designed to log the event in real-time or delayed/aggregated?	The accuracy of temporal measurement is crucial for sequence analysis as it forms the basis of creating the correct sequence. For example, the system logs event A in real-time but logs event B on the hour.
Consistency of data	Has the data been recorded in the same way, such as format and categorization, throughout the study?	Inconsistency can lead to incorrect analysis results. For example, event A and event B are recorded as two separate events at first but then recorded as the same event A later due to a change of system definition.

The processing of raw digital trace data requires a strong theoretical foundation to address several challenges. The key decision to make is how to define the events and create a sequence of events. Researchers are required to define the ontology of discrete “events” (Hui 2016). Langley (1999) discusses difficulties in defining events with process data concerning elements that need to be included as part of an event. The conceptualizing of events should take both the theoretical needs and empirical availability into consideration. Theoretically, the ontology should align with the research interests and should assist in answering the research questions. Empirically, the ontology should have a connection with the data, either directly or indirectly. Normally, the process data should contain “what happened and who did what when” (Langley 1999). Similarly, Pentland (1999) discusses several key features of a story when developing process theory through narrative: (1) sequential structure; (2) focal actor(s); (3) narrative voice; (4) evaluative context; and (5) other indicators of context.

When developing an ontology, researchers' interference needs to be minimized so as not to nudge the coded empirical data into the domain of the actual. Thus, it should be as straightforward as possible while maintaining theoretical relevance. For example, Gaskin et al. (2014) develop an ontology with seven different elements (activity type, actor configuration, location, tool modality, tool affordance, artifact type, and dataflow) to conceptualize a design event using digital tools.

Another key task with event ontology is to identify the proper unit and level of analysis. As the digital trace data collected often have a fine granularity, researchers need to decide whether to aggregate the data to a higher level via abstraction. The level of analysis should be chosen per the level of events that constitute the outcome under study. For our illustrative example, the research question under consideration is concerned with the collaboration pattern; therefore, the unit of events for the analysis is action on a Wikipedia article.

While collecting and processing digital trace data, researchers should also pay close attention to potential issues related to data validity and reliability. Howison et al. (2011) identify several issues, particularly issues related to the information systems that generate the digital trace data, which can affect the validity and reliability of the digital trace data. They also provide recommendations to deal with these issues. In short, researchers should always make a great effort to have a thorough understanding of how the system is designed and used and how the data is measured and recorded by the system.

The Illustrative Example

As the goal of the illustrative example is to understand collaboration patterns for creating featured articles, we collected records on actions related to edits of articles and interactions among actors. Wikipedia automatically records all types of user activities in its log system that can be retrieved through the official API[4]. We developed a Python script to download all the editing activities provided by the API. For each Wikipedia article, there are two types of pages. The first is the main

[4] Such sequence-centric computational models, we believe, can be useful not only to qualitative scholars interested in process studies, but also to those who are interested in the computational method in general as a complementary approach to network-centric computational models that have gained popularity among IS scholars. While the network-centric approach focuses on discovering underlying latent structural features that affect social phenomena, the sequence-centric approach focuses on discovering latent temporal features that underpin changes in social entities. We believe this would be particularly helpful for qualitative scholars who are interested in conducting process studies.

page, which contains the actual content for readers. The other is the talk page, where contributors or readers may make comments. We collected revision histories for both pages over the first 90 days[5] of each article's existence to understand the process of creating an article. According to Wikipedia's API policy, each article's revision history tracks all changes: a change made to the main page shows an edit was made to the article content, and a change made to the talk page is shown as a comment on the discussion of the content. As shown in Table 3, we combined the main and talk pages' revision histories to create one single timeline based on the timestamp.

Table 3 Example of collected data.

Article Title	Timestamp	Contributor	Action
1911 in Iran	2015-05-21 20:49:54 UTC	F4fluids	Edit
1911 in Iran	2015-05-21 21:37:43 UTC	Wgolf	Edit
1911 in Iran	2015-05-22 14:02:16 UTC	F4fluids	Edit
1911 in Iran	2015-06-10 05:46:43 UTC	Ser Amantio di Nicolao	Talk

We collected data from all 4,144 featured articles[6] in English Wikipedia, which make up less than 0.1% of the 5 million English Wikipedia articles. There was no restriction on the featured articles' topic, so these featured articles were selected from 30 different major topics through the nomination and review process[7]. To compare, we also collected the same set of data from a random sample of 4,000 English non-featured articles. Following suggestions by Langley (1999) and Pentland (1999), we developed the coding scheme using actor and action (see Table 4). This coding scheme reflects the ontology of our sequence analytics for the study. We classified actors as article creators or non-creators. Such classification was driven by our research interests to

[5] We also collected 6 months' data. The results are similar for the first 90 days, 120 days, and 180 days. Therefore, we present the 90 days results for the simplicity of the case study.

[6] The time the data was collected was in Oct. 2016.

[7] Please refer to https://en.wikipedia.org/wiki/Wikipedia:Featured_articles for a full list of featured articles and the topics.

investigate the collaboration at the early stage of knowledge creation when the creator initiates a new Wikipedia article, and non-creator contributors make their contributions to it later.

Table 4 Definition of collaboration events.

Sequence	Order ID	The order of the event in the editing process based on the timestamp
Actor	Creator (CR)	The initial contributor who created the Wikipedia article
	Non-Creator (NC)	Any contributor other than the article creator
Action	Edits (ED)	Make a change to the content of the article
	Talk (TK)	Post a comment on the talk page of the article

Action is comparable to the mixture of identifiable voice and additional information, as it is the recordable form representing the actor's viewpoint from the digital trace data. Then, we classified actions as edit or talk, which are the two primary types of actions that a Wikipedia contributor can make. There are thus four types of events: creator-edit, creator-talk, non-creator-edit, and non-creator-talk[8].

We then built the sequences of four types of events based on their timestamp. Table 5 shows how the data shown in Table 3 can be coded based on our ontology[9].

Table 5 Coding results.

Article Title	Article ID	Timestamp	Order ID	Contributor	Action	Coding
1911 in Iran	1	2015-05-21 20:49:54 UTC	2	CR	ED	CRED
1911 in Iran	1	2015-05-21 21:37:43 UTC	3	NC	ED	NCED
1911 in Iran	1	2015-05-22 14:02:16 UTC	4	NC	ED	NCED

[8] We also kept the events simple for illustrative purposes.

[9] Assume F4fluids is the article creator.

1911 in Iran	1	2015-06-10 05:46:43 UTC	5	NC	TK	NCTK
The example sequence created based on the sample data is: CRED-CRED-NCED-NCED-NCTK						

Step 2: The Computation of Sequence Analytics

Once the collected data have been processed into a proper format, we perform computational steps to discover latent patterns. Then we identify primitive patterns that represented groups of closely related latent patterns. These primitive patterns are then used to approximate the *actual* events (Wynn and Williams 2012) or routines in this study context.

Step 2.1 Discovering the Latent Patterns

Empirical observations may differ from actual events because of the variations caused by the specific context where the event takes place and using observations and perceptions (Wynn and Williams 2012). By discovering the latent patterns from multiple sequences collected from different instances at different times, we can overcome such issues. The discovery of the latent patterns can be achieved with computational approaches like sequential pattern mining.

Sequential pattern mining was first introduced by Agrawal and Srikant (1995). The primary objective of sequential pattern mining is to find frequent sub-sequences in a large dataset (Fournier-Viger et al. 2017). Frequent sub-sequences are segments of events that appear statistically more frequently than others across different sequences. In other words, the events in the sub-sequence are not coexisting and are ordered sequentially by coincidence. Therefore, frequent sub-sequences represent latent patterns across different contexts. Figure 2 shows how the original sequence can be seen as a series of frequent sub-sequences, which are denoted by upper case letters. We would like to highlight that a frequent sub-sequence is not determined by

the similarities between adjacent events but by the occurrences of ordered events across different sequences. In this example, we use the same lower-case letters to denote the events that appeared in similar sub-sequences (e.g., a1 may differ significantly from a2, while both come from A). There are four observed variants of A, including A1 and A'1 from sequence P1, and A2 and A'2 from sequence P2. It is often much more complex than the examples we show here.

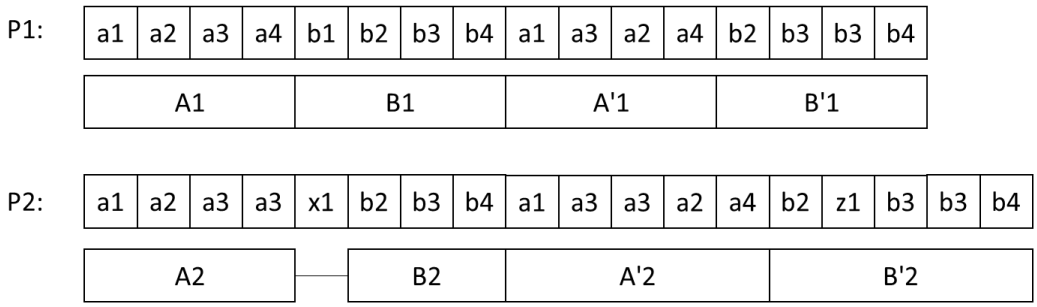

Figure 2 Latent patterns discovered from the empirical process.

Since the invention of sequential pattern mining, various algorithms have been developed. Sequential pattern mining is essentially an enumeration problem that should always have "a single correct answer" with the same criteria, regardless of which algorithms to use (Fournier-Viger et al. 2017). These algorithms are based on deterministic algorithms that only differ by their search strategies, so that some algorithms may be more efficient than others.

To effectively apply those algorithms, many criteria can be set to limit the search strategy. Generally speaking, two criteria are applicable to most algorithms and also most relevant to process studies: occurrence frequency and length. Occurrence frequency, commonly referred to as support, is the most important criterion. Minimal support specifies the minimal percentage (or absolute number) of sequences in which the particular sub-sequences should be found. For example, if one dataset contains ten sequences and the minimal support is set to 70%, then a sub-sequence needs to appear in at least seven sequences to be considered a frequent sub-

sequence. Max length specifies the maximum number of events in a sub-sequence. For example, if the max length is set to 6, only sub-sequences with a length from 2 to 6 will be selected.

Both theoretical considerations and computational constraints should drive the decision regarding minimal support and max length. On the one hand, how "repetitive and recognizable" a routine should be is not decisive, particularly when the dataset is large. Arguably, minimal support mainly concerns the repetitiveness so that the larger the dataset is, the lower the minimal support may be set. Max length mainly concerns the recognizableness so that the longer the whole sequences are, the greater the max length may be set. On the other hand, lowering the minimal support and increasing the max length would significantly increase computing time. This may pose a significant constraint when dealing with very large datasets with long sequences. Therefore, it is usually a "trial-and-error" practice to balance the theoretical consideration and computational constraint. Traditional routine studies can often offer some inspirations to set up the parameters. For example, Pentland and Rueter (1994) report that the average length of their routines is 9.57 with a standard deviation of 2.77. In another study, Pentland et al. (2010) report average length around 10 and 14 for two different routines. Based on our experience, the complexity of the task, stability of the organization, and heterogeneity of actors can all contribute to the potential length and support, such as routines in virtual communities often shorter than in traditional organizations. Generally, we suggest starting with lower constraints (e.g., minimal support 20% and max length 8), which may require longer computation time but give a better overview, and then gradually adjusting them. APPENDIX A provides a summary of commonly used sequential pattern mining algorithms and other important search criteria.

The Illustrative Example

We used TraMineR (Gabadinho et al. 2011), an R package specially designed for sequence analysis in the social sciences. It has been used by many academic researchers in different

disciplines, including IS (Lindberg et al. 2016). We used *seqefsub,* a function of TraMineR, to identify frequent sub-sequences with a set of constraints. This function is based on a prefix-tree-based search algorithm (Ritschard et al. 2013). APPENDIX B discusses the consideration of key parameters that we used in our example.

To understand different collaboration patterns from both micro-level (i.e., between different articles) and macro-level (i.e., across different parts of the Wikipedia community), we further broke down our data into five three-year periods (2001 to end of 2003; 2004 to end of 2006; 2007 to end of 2009; 2010 to end of 2012; 2013 to end of 2015)[10]. Table 6 shows the number of featured and non-featured articles in five time periods. The break-down is based on existing studies on Wikipedia growth in terms of the number of articles[11] and active editors (Halfaker et al. 2012). We then performed sequential pattern mining analysis for each period using the parameter specified in APPENDIX B. Table 7 shows an example set of results for featured articles created during 2001-2003. The sub-sequence ((CRED)-(NCED)) can be found in 60.13% of the featured articles created in that period. Similarly, ((NCED)-(NCED)) can be found in 53.50% of the featured articles. We present more detailed results in APPENDIX C.

Table 6. Summary of Wikipedia articles from 2001 to 2015

Period	Featured Article	Non-featured Article
2001-2003	1402	155
2004-2006	1553	1021
2007-2009	716	1289
2010-2012	289	800

[10] Although the number of featured and non-featured articles in each year can be uneven, in the three-year time period, there are statistically sufficient samples for the comparison.
[11] https://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia_extended_growth

2013-2015	171	672

Table 7 An example of the sequential pattern mining result.

Featured Article	
Sub-Sequence	Percentage
(CRED)-(NCED)	60.13%
(NCED)-(NCED)	53.50%
(CRED)-(NCED)-(NCED)	38.45%
(NCED)-(NCED)-(NCED)	36.80%
(CRED)-(CRED)	28.25%

Step 2.2: Identifying the Primitive Patterns

Sequential pattern mining helps us to discover those latent patterns. However, not every latent pattern should be seen to be a demi-regularity due to embeddedness, minor variations, overlapping, and incompleteness. Therefore, we need to further process the latent patterns through clustering analysis to identify the primitive patterns. Identifying the primitive patterns is the second step to move from the *empirical* domain to the *actual* domain. These patterns are a set of sequential patterns with a high level of abstraction that serves as primitive forms of a much larger number of latent sequential patterns. Primitive patterns constitute emergent *demi-regularities* in our efforts to discover the generative mechanisms.

Primitive patterns can be identified through clustering analysis. A cluster represents a group of sub-sequences, the latent patterns identified in the previous step, that share similar patterns, allowing researchers to compare and contrast different latent patterns within and across different clusters. Such cross-comparisons of latent patterns across clusters, through a priori or an emergent theoretical lens, help scholars to make sense of the empirical findings. Figure 3 shows an example of how we can identify primitive patterns through clustering. Identifying clusters

requires two major computational operations: creating a distance matrix based on sequence similarity/dissimilarity, and creating clusters based on the distance matrix[12] (MacIndoe and Abbott 2004).

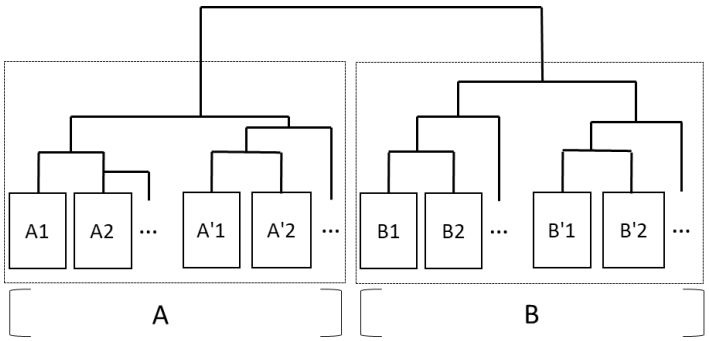

Figure 3 Primitive patterns (A and B) identified through clustering analysis.

Measuring the dissimilarity between sequences is the most common and significant starting point for sequence clustering. The dissimilarity measure can be formulated through a mathematical definition of distance between sequences. There are three main methods to calculate the distance between sequences: distances between probability distributions, distance based on counts of common attributes, and optimal matching (Studer and Ritschard 2014). Generally, we recommend optimal matching, as it is the most commonly used one in social science for its simplicity and ease of use (Studer and Ritschard 2016). The results can vary with different methods, and we discuss the different measurements in detail in APPENDIX D.

In the OM approach, the key parameter is the substitution matrix, which defines the substitution cost between each event type. While it is ideal to have the substitution matrix derived from prior

[12] For clustering sequential data, there are also less common non-distance based methods based on feature extraction or hidden Markov models. However, these methods generally would cause loss of information (Xu and Wunsch 2005) so we do not include them in our discussion.

theory, it is impractical if there are no prior studies. When theory-driven cost is unavailable, the researcher can choose to use either constant cost, which considers every substitution to be equal, or data-driven cost, which is derived from empirical data based on the observed probability of change. Whenever the data-driven cost is available, it should be preferred, as it better reflects the actual relationship between events. Wilson (2006) suggests using low substitution cost when the actual is unknown. Studer and Ritschard (2016) show that the data-driven substitution cost has a limited effect. Similarly, in another study, Gauthier et al. (2009) show great similarity between the final results of constant cost and other empirically grounded cost strategies.

After the distance matrix had been generated, we apply clustering techniques to create the clusters. We recommend hierarchical clustering over other popular clustering techniques, such as partitional clustering, as it generally provides more robust and deterministic results while allowing researchers to visually oversee the clustering (Mihaescu et al. 2009; Reddy and Vinzamuri 2013). In APPENDIX E, we provide a detailed discussion about our recommendation of hierarchical clustering and the key considerations in hierarchical clustering.

The key parameter of hierarchical clustering is the linkage criteria to join clusters in the hierarchy. There are several common linkage criteria used in hierarchical clustering: single linkage, complete linkage, average linkage, and Ward's linkage. The choice of linkage criteria can affect the formation of clusters. In other words, the transformation of the data can be altered by the researchers' specifications to the clustering algorithms. We recommend Ward's linkage over the others because, unlike other criteria that evaluate pairs of sequences between clusters, Ward's linkage evaluates pairs of sequences within clusters (Cornwell 2015). That is to say, Ward's linkage emphasizes the similarity of sequences within a cluster, while others emphasize the dissimilarity of sequences across clusters. Therefore, the Ward method better served our purpose of identifying demi-regularities through clustering sub-sequences. However, we also note that the

Ward method has its limitations. For example, it tends to generate clusters with a similar number of sequences, and it does not perform well where there is a severe outlier issue (Cornwell 2015; Ketchen Jr and Shook 1996). Given the impact of linkage choice, we also recommend researchers try other criteria and visually inspect the clustering results. Such a clustering process can be iteratively calibrated with the retroductive reasoning in the next step.

The primitive patterns are demi-regularities that can then represent actual happenings that persist across different instances. These primitive patterns should be interpretable and theoretically meaningful. The purpose of primitive patterns is not to simply remove unnecessary or uncommon variations from the data but to go beyond individual observations and focus on the underlying process that can sometimes be varied and changed because of the specific contexts where it is performed.

The Illustrative Example

To identify further primitive patterns, we conducted a clustering analysis on frequent sub-sequences discovered previously. The distances between sequences are calculated with the function *seqdist* in TraMinR. We used optimal matching (OM) as the measurement method. For the substitution cost (and based on the previous discussion), we set it as "constant," as we could not find the prior research necessary for assigning a theory-based cost[13].

Next, since we identified only 172 sub-sequences with a maximum 8-event length from step 2, there is hardly any difference in computational efficiency between *k*-mean clustering and hierarchical clustering. We applied hierarchical clustering for the reasons we discussed above (Figure 4). We used the R built-in function *agnes*. While there are different quantitative methods

[13] Please also refer to APPENDIX D for further discussion on the decision of substitution cost.

(such as the elbow method and the silhouette method) to help determine the number of clusters (Ketchen Jr and Shook 1996), there is no universal choice. This is also not an issue specific to sequence analytics. The choice is often determined not only by quantitative numbers but also by theoretical standpoints and interpretability. It is also an iterative process, testing the different numbers of clusters through different criteria and methods.

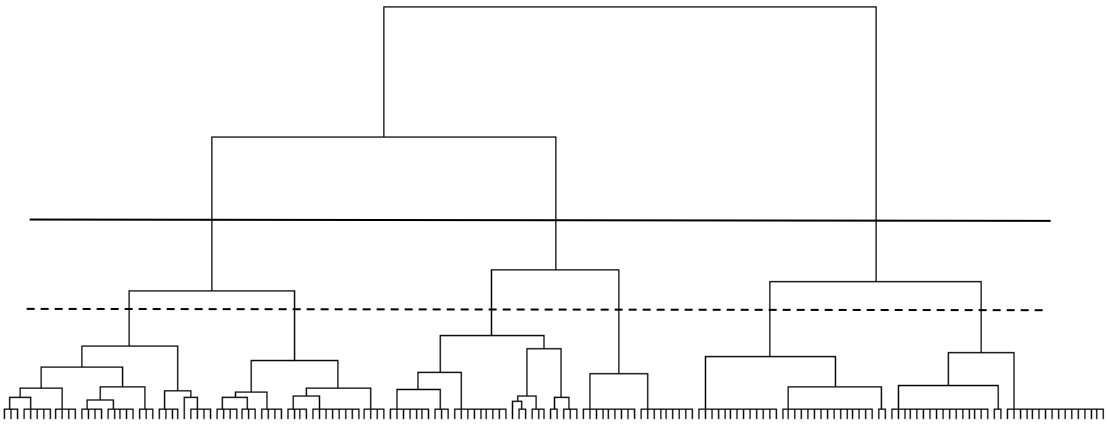

Figure 4 Dendrogram of clustering analysis[14]

A study by Ketchen and Shook (1996) on the application of clustering analysis in strategy research suggests that combined use of multiple methods can improve confidence in the results. Here, the choice is mainly between 3 clusters and 6 clusters. We performed an analysis with both choices and checked the results through manual inspection. We found that the 3-cluster model produces a more reliable interpretation, as shown in APPENDIX F. Therefore, we identified three primitive patterns: non-creator dominated sequences (cluster 1), mixed short sequences (cluster 2), and creator-dominated sequences (cluster 3). All the sub-sequences we discovered can be seen as

[14] The solid line represents cutoff for 3 clusters and the dash line represents cutoff for 6 clusters.

some form of variation of these three generic meta-sequences. The process of article creation can be characterized by a combination of the three generic meta-sequences.

In Figures 5, 6, and 7, we summarize the highest probability (the probability of the most frequently occurring sequence) of each meta-sequence appearing for articles created in different periods[15].

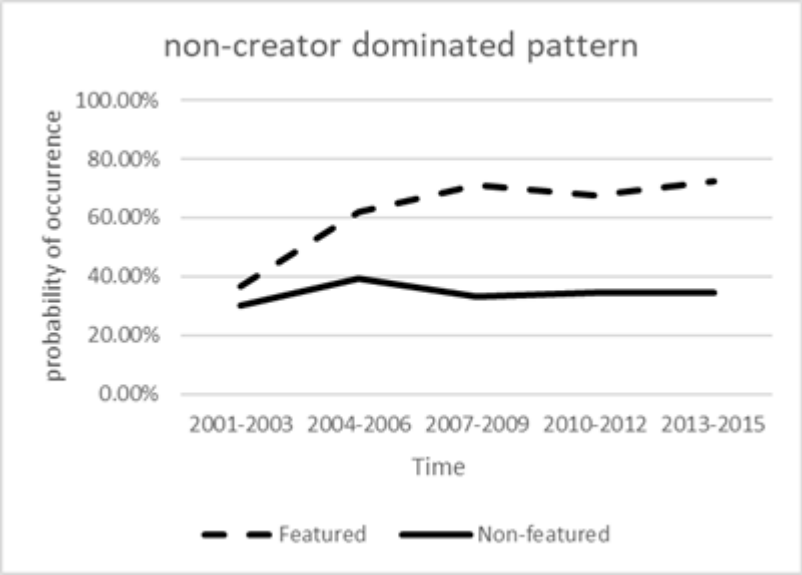

Figure 5 Non-creator-dominated pattern changes over time.

[15] For the purpose of illustration, we set occurrence to 30% if the pattern is not detected as a frequent sub-sequence. The actual occurrence will be lower than 30%.

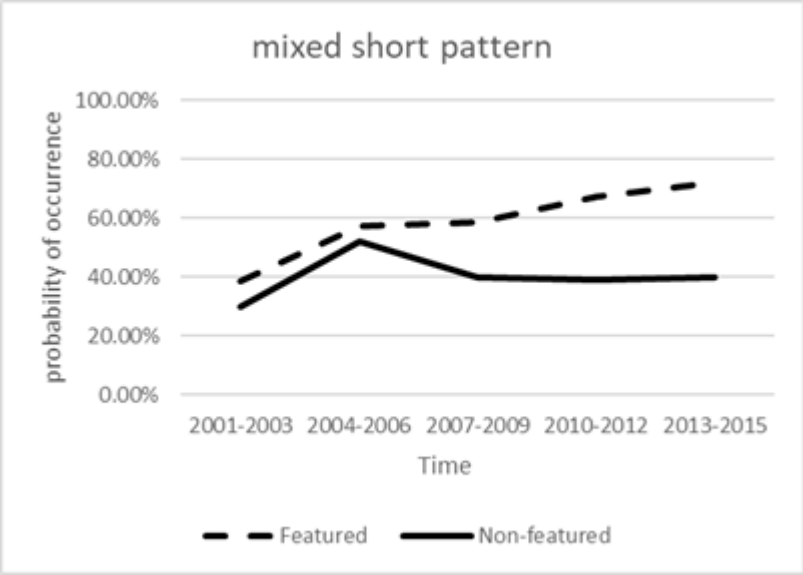

Figure 6 Mixed short pattern changes over time.

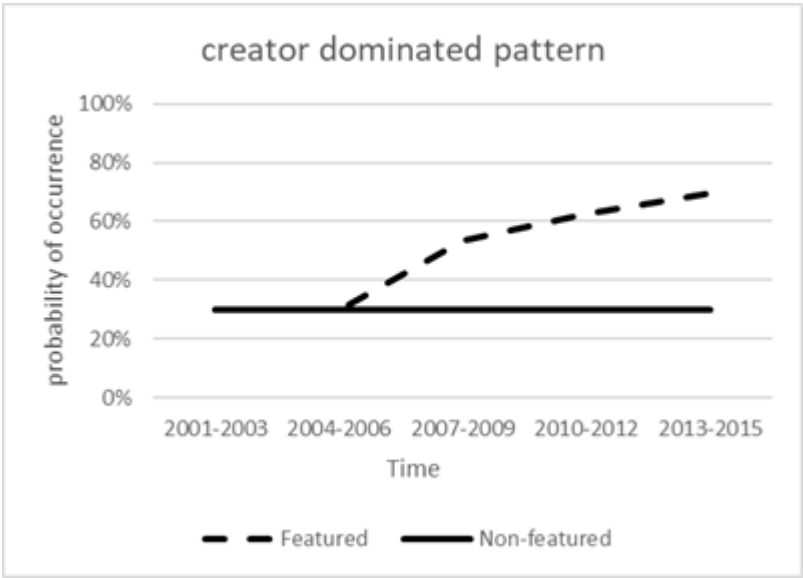

Figure 7 Creator dominated pattern changes over time.

Step 3: Interpreting the Results

In this step, using primitive patterns identified through sequence analytics and their relationships, researchers seek to explore the underlying generative mechanisms through retroduction. The results from sequence analytics require context-dependent interpretations through retroductive reasoning to identify the real mechanisms from possible alternative explanations. The purpose of retroduction is to identify possible mechanisms that can produce the observed events. However, since the critical realist acknowledges that the underlying mechanisms are not directly observable and that the observed events are usually constrained by human perception, there may be multiple possible mechanisms that may generate the observed events (Wynn and Williams 2012). Thus, we need to identify plausible candidate mechanisms that "if they existed, would generate or cause that which is to be explained" (Mingers 2004), and then select the one with the best explanatory power. The existing literature can offer a good starting point to identify these candidate mechanisms. In APPENDIX G, we provide several examples of how existing theories may help us derive potential generative mechanisms.

The Illustrative Example

Based on our sequential pattern mining analysis and clustering analysis, we identified three primitive patterns in the online collaboration on Wikipedia. From Figures 5 to 7, we can see a steady upward trend for all three meta-sequences in the featured article creation processes that are not seen in the non-featured article creation ones. While this change partly comes from the increased total number of edits, the reason is probably more complex than that. While it is commonly believed that Wikipedia is built by "the wisdom of the crowd" (Aaltonen and Seiler 2015; Kittur and Kraut 2008; Ransbotham and Kane 2011), the dramatic increase in the creator dominated pattern in the creation of featured articles belies such an assumption.

The results of our analysis show that the collaboration patterns are quite different for the featured and non-featured articles, and the differences become even more pronounced over time, particularly after 2006. During the early years of Wikipedia (i.e., 2001 to 2003), the collaboration patterns are fairly similar between the featured and non-featured articles. However, the featured articles created during this period have more non-creator continuous editing than non-featured articles. At that time, Wikipedia was still a little-known website, and usually, the contributors were technophiles[16]. To some extent, they were more homogenous than the contributors in later years, and they were more likely to share similar visions and beliefs. The majority of original article creators did not make significant efforts in continuous editing in either type of article (less than the 30% threshold).

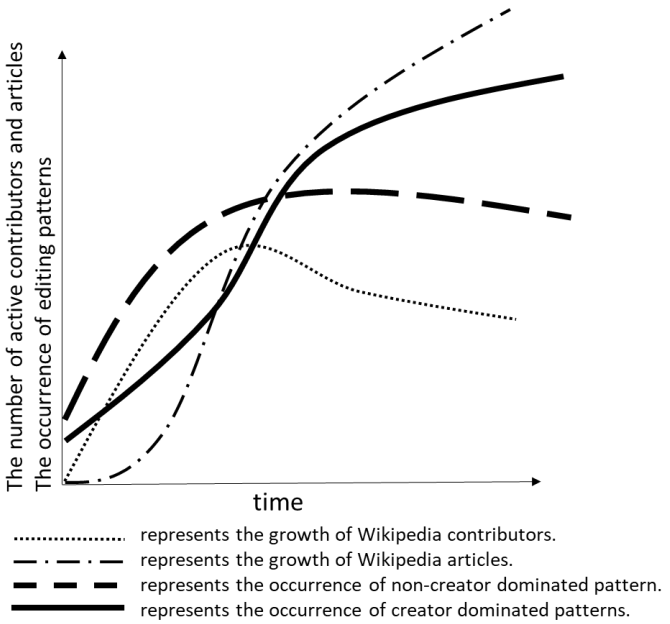

Figure 8 Change of collaboration patterns of Wikipedia[17]

After 2004, however, as Wikipedia started growing rapidly and was drawing an increasing number of editors worldwide, we see a change in the collaboration patterns. For the articles created

[16] https://meta.wikimedia.org/wiki/Research:Wikipedia_Editors_Survey_2011_April

[17] The growth of Wikipedia is based on existing studies we referred to in an earlier section. This figure is mainly for illustration of approximate trends.

between 2004 and 2006, continuous editing, from both the creators and non-creators, nearly doubled for both featured and non-featured articles. Since 2007, as Wikipedia's growth slowed, the collaboration patterns changed once again. However, the changes differed from each other depending on whether the articles were featured or non-featured. There was another large increase in creator-dominated patterns for featured articles created between 2007 and the end of 2009. During the same period, the collaboration patterns for non-featured articles remained roughly the same. There was even a drop in non-creator dominated edits. Since 2010, as the decline of Wikipedia has stabilized, the change in collaboration patterns has continued. In particular, the increase in creator-dominated activity continues with featured articles, while non-creator-dominated actions have stayed the same for both featured and non-featured articles (see Figure 8).

Looking at our results through the four modes of change proposed by Van de Ven and Poole (1995), we propose a life cycle as shown in Figure 9 as a process model of online collaboration's dynamic evolution. The life cycle perspective is appropriate in this context as the changes in collaboration patterns of Wikipedia have to do with the changes of users over time as the community grows. Our model extends the stage model proposed by Iriberri and Leroy (2009) suggest that an online community may go through its life cycle: inception, creation, growth, maturity, and death. The latter four stages are relevant to our framework as the inception of a vision cannot be captured through the empirical data.

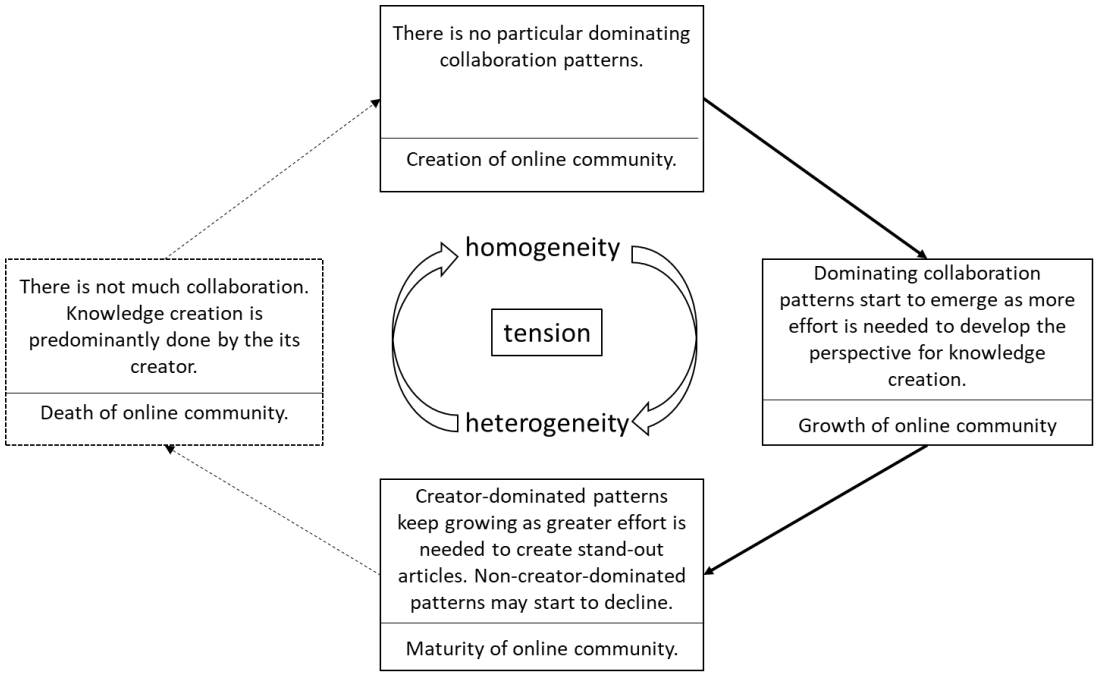

Figure 9 The life cycle of online collaboration and corresponding stages of the online community.

We use the dash-line box to highlight the potential collaboration patterns as the community decreases. Looking at each stage as a micro-online community where people collaborate around a particular topic, our analysis shows how creator-dominated and non-creator-dominated collaboration patterns can grow and decline as the community changes. Our model provides an underlying generative mechanism – the ongoing tension between the online community's homogeneity and heterogeneity – that produces the distinct stages as the community grows. As the community grows, norms, policies, roles, and procedures become institutionalized (Bagozzi and Dholakia 2002; Fleming and Waguespack 2007; Murray and O'Mahony 2007; O'Mahony and Ferraro 2007; Ransbotham and Kane 2011). At the same time, more heterogeneous members join the community because of the online community's fluid nature (Faraj et al. 2011). As a result, tension rises between the original homogenous members and the newly joined more heterogeneous members (Eaton et al. 2011; Zhang et al. 2016). When tension is not properly managed (as can be seen in favoring old members), new editors are discouraged and deterred

from contributing. Simultaneously, the old editors become experienced in creating good articles following the Wikipedia standards and policies. Therefore, we see a steady increase in creator dominated patterns in articles created in the later years.

Step 4: Validating the finding

Once scholars have identified the generative mechanism through the retroductive process, the conclusion is supportive rather than conclusive (Mingers 2004). Therefore, we need to validate the identified mechanisms. This can be done by assessing and eliminating alternative explanations. One commonly used approach is empirical corroboration, which corroborates potential mechanisms with empirical evidence (Wynn and Williams 2012). For example, Bygstad (2010) compared the explanatory power of an identified mechanism with the explanatory powers of other possible alternatives in the empirical setting to eliminate the alternatives. Williams and Karahanna (2013) sought corroboration through repeated confirmation, including confirmation from multiple participants and cases. Another validation approach is triangulation, in which multi-methods and/or data sources can be utilized to confirm the validity of the proposed mechanism. For example, Zachariadis et al. (2013) and Volkoff et al. (2007) leveraged data obtained from different sources to confirm their mechanisms.

Advances in computational science also provide potential ways of validating the findings. For example, machine learning and artificial intelligence share a similar philosophy. By analyzing a set of data, the computer tries to learn the classifier/reasoning with the best predictive power in other data sets. Thus, it is theoretically possible that scholars can design a “classifier” based on the mechanism identified through sequence analytics and test its explanatory power using digital trace data from other cases. In reality, there is no universal way to validate the identified

mechanism. Validation is often seen not as a separate step but more as part of the iterative retroductive process (Fletcher 2017; Zachariadis et al. 2013).

The Illustrative Example

The validation of the generative mechanism was achieved in several different ways. First, it was done as part of the iterative retroductive process in step 4, where we selected the life cycle over the other potential modes proposed by Van de Ven and Poole (1995). The other three modes do not provide comparable explanatory power to the life cycle model. The patterns we see from our data do not match any possible patterns generated by the other three.

Second, we sought confirmation from empirical corroboration and triangulation. Besides collaboration patterns, we also expected a change in the editors' characteristics, which was not observed in our data but should be caused by the life cycle mechanism as the community goes through different stages. Such a change in characteristics then leads to a change in collaboration patterns. To confirm our expectation, we utilized other data sources, such as discussions between editors on Wikipedia, official reports, and other studies. For example, there are intense discussions among users, surveys, and official projects by Wikipedia that try to address editor engagement and retention. A report suggests that editing on Wikipedia by the creator to create a good article has now become newcomer-unfriendly [18]. Another study found that potential new editors are driven away and that the newcomers encounter difficulties getting into the inner circle of Wikipedia (Halfaker et al. 2012).

[18] Some examples include: https://commons.wikimedia.org/wiki/File:Editor_Survey_Report_-_April_2011.pdf; https://en.wikipedia.org/wiki/Wikipedia:Please_do_not_bite_the_newcomers; https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Editor_Retention;

DISCUSSION

Our primary contribution is a novel methodological framework for applying computational sequence analytics to build a process theory of routines based on critical realism. Our framework provides a philosophical foundation and an operational prescription for IS scholars to investigate and theorize changes in routines using digital trace data. With our framework, scholars can use computational tools to detect and visualize latent temporal patterns of routines. Table 8 summarizes key decisions on methodological principles, potential issues, and recommendations for each step of the framework. Our position is that the discovery of such latent patterns allows scholars to go beyond the observed patterns' variance and focus on the underlying generative mechanisms and social forces that produce those observe patterns. Our framework offers a powerful means of leveraging a large volume of digital trace data of virtual activities produced by the information infrastructure in studying routines in organizing.

Table 8 Summary of the sequence analytics framework

Steps	Decision	Methodological principle	Potential issues	Recommendations
Data preparation	What is the best way to define the events and create the sequence with digital trace data generated by the system?	Determine the level of events and the constituent parts of the structure when developing the event ontology and creating the sequence.	The validity and reliability of the digital trace data.	Focus on the analysis level and parts that are most relevant for the research objectives.
Sequence analytics - Discovering latent patterns	What is the best way to extract common events from digital trace data? What are the most appropriate computational techniques?	Explicate the event, structure, and context through sequence mining.	Unreliable results from inappropriate use of computational techniques.	Become familiar with the underlying assumptions and requirements of different computational approaches.
Sequence analytics – Identifying primitive patterns	What is the best way to further abstract common event sequences into demi-regularities?	Explicate the event, structure, and context through classifications.	The level of further abstraction may vary, depending on the specification of computation techniques, such as the number of clusters.	Establish a balance between interpretability and the number of demi-regularities. Provide triangulation with other methods.
Interpretation	What is the most likely explanation for the identified demi-regularities?	Identify plausible causal mechanisms.	Over retroduction leads to a generative mechanism that has no or a weak linkage with the demi-regularities.	Given that it is a “thought trial” (Weick 1989) that needs to be done in an iterative manner know the possibility of alternative mechanisms.
Validation	Does the proposed mechanism indeed have the better causal power to explain the observed experience?	Find empirical corroboration to demonstrate the proposed mechanism.	Context-specific validation may lose the generality of the findings.	Apply the proposed mechanism in other settings. Compare with other studies.

In particular, we propose a critical realist approach to address the multi-layered nature of the routines by treating the digital trace data, which is large in volume but lean in context, as empirical observations. Empirical observations collected through digital trace data represent varieties of routines, where routines are generated by the underlying mechanisms and just realizations of many possible routines that can be generated, with possible variances and noises introduced during the realization in specific contexts. In the past, scholars conducting critical realistic studies had to make great efforts to collect and analyze data, often manually, to move from the empirical domain to the real domain. Such an approach is extremely challenging to study non-traditional forms of organizing due to the immense variation because of the diversity of the settings and many unobservable factors. Furthermore, moving from the empirical observation to underlying mechanisms becomes even more important when a large amount of digital trace data becomes available and can depict the latent patterns. Therefore, our framework offers scholars a means to uncover generative mechanisms from digital trace data collected even in non-traditional settings. With primitive patterns identified by sequence analytics, researchers can delve below the surface of casual relationships and focus on the underlying mechanisms.

We show how various computational techniques may well align with the ontological assumptions of critical realism. The underlying assumption of sequence analytics is that the input sequences are not randomly generated, and there are latent stochastic relationships among the events in the sequence. The generative mechanisms shape these latent relationships. Therefore, some sequential patterns may appear more frequently than others, and sequence analytics offer a more efficient way to identify those patterns, which can be considered an approximation process of the observable phenomena. In our framework, we suggest a two-stage approach with sequential mining and clustering. Researchers would analyze a stratified ontological view of critical realism to approximate the actual from the empirical observation. In the first stage, the sequential pattern mining approach does not transform the observation, as it merely identifies the sub-sequences

that appear more often across different processes. In the second stage, the clustering analysis aims to classify these sub-sequences based on similarity. Each group represents a demi-regularity produced by the generative mechanism, and each group member is a realized observation. As the clustering is done by computing the similarity, the result's validity can be significantly affected by the alignment between the true relationship and algorithm assumptions. This is especially important because the true relationship, as the transition probability in the Markov process[19], is often unknown or difficult to obtain. For example, the distance between two different sequences may change due to the different substitution costs specified in the optimal matching algorithm, leading to different clustering results. Similarly, the clusters' memberships may change dramatically due to the randomness of initial centroids and the number of clusters in k-mean clustering. It assumes that the observations that share a similar distance to the centroids should be considered in the same group.

Furthermore, our framework also makes a broader contribution to critical realistic research in IS. As there is so far no unitary framework and methodology for applying critical realism in research (Archer et al. 2016), it is problematic for IS scholars to adopt this philosophical way of thinking. Our study shows an alignment between critical realism and sequence analytics with digital trace data to study the social phenomenon. We believe such an alignment can also be applied to other research applying pattern recognition analytics in big data studies. This is especially true when the dramatically increased data size provides diminishing benefits to traditional quantitative approaches, such as statistics.

[19] Here we use the Markov process only as a representative stochastic process to give an example of how latent relationships may be translated into a computational model specification. The interdependency between events can be more complex in many cases.

Our framework also addresses one of the weaknesses in the way computational methods are applied in social science. Namely, computational approaches are commonly criticized for having a predictive capability but lacking explanatory capability. Muller et al. (2016) discussed how the use of computational analysis in social sciences generally follows either an explanatory or a predictive approach, where explanatory studies mainly rely on statistical analysis, and predictive studies primarily use data-driven approaches like data mining. Particularly, they criticized that data mining "is not quite in line with the traditional scientific method" because science should be about finding explanations rather than finding patterns (Muller et al. 2016, p. 291). Such an issue is often related to the epistemological perspective of positivism in most data-driven predictive studies. Our framework answers such critics by showing that computational tools, such as sequential pattern mining and clustering in our case, can fit well with critical realism as a legitimate method of scientific inquiry. Our methodological framework aims to offer causal generative mechanisms that produce the empirical patterns observed through computational tools. Computational techniques process the empirical observations from the big data into primitive patterns representing actual happenings that humans may not observe. While such computationally generated primitive patterns may not necessarily be true actual happenings, they depict what the actual happening is likely to be. We consider these primitive patterns as "the computational actual." While studies using computational analysis as a predictive approach usually stop at this point, with computational results as the conclusion, studies following our framework do not end here and are followed by further retroductive examination to find the underlying mechanisms that cause the computational actual identified by computational analysis.

Table 9 compares this study with other similar studies on routines from contribution, philosophical paradigm, computational methods, and data perspectives. While most studies focus on modeling routines and develop theories based on interpretivism and positivism, this paper provides a systematic approach to identify and theorize generative mechanisms with regards to routines

based on critical realism. The majority of the studies in Table 9 adopt network modeling to represent the sequence of actions for theorizing routines.

Table 9. Comparison among recent routine studies based on computational methods

Studies	Contribution	Philosophical paradigm	Computational methods	Data
Pentland et al. 2010	To conceptualize patterns as network of actions and compute network metrics so as to allow researchers to investigata generative mechanisms of routines.	Interpretivism	Network analysis	Digital trace data (workflow event logs)
Gaskin et al. 2014	To measure the variations among sociomaterial routines, which can be used together with qualitative data as a mix-method approach to study organizational routines.	Interpretivism	Lexicon modeling and sequence analysis	Qualitative field data
Vaast et al. 2017	To discover patterns of interdependence as network motifs so as to investigate how social media use affords new forms of collective engagement.	Critical realism	Clustering and network motif analysis	Digital trace data (tweets)
Schecter et al. 2018	To identify patterns of interactions so as to test if they are associated with process quality indicators, coordination and information sharing.	Positivism	Relational event modeling	Digital trace data (system logs of online team game)
Goh and Pentland 2019	To compute the quantitative metrics, such as complexity, of the narrative network, and then combine with qualitative data so as to theorize the dynamics of routines.	Interpretivism	Network analysis	Digital trace data (scrum sheet) and field data (archive, interviews, observation).
Griswold et al. 2020	To propose process mining as a computational tool for theorizing change processes in organizations.	Positivism / Interpretivism	Process mining	Digital trace data
Our framework	To provide a computational methodological framework to analyze digital trace data grounded with stratified ontology from critical realism, which can be used to discover routines and identify underlying generative mechanisms	Critical realism	Sequence analytics, clustering methods	Digital trace data

While network modeling has an advantage in analyzing structural characteristics of sequences of actions, it tends to analyze sequential relationships as interdependences between adjacent events in the network. The relationships among non-adjacent events in a process cannot be fully captured by this method. The sequence analytics proposed in this paper has an advantage in identifying primitive patterns and analyzing their evolution paths. Viewing digital trace data as empirical obvervations in stratified ontology, our framework can also take full advantage of scalability for big data sets. The digital trace data has also been utilized and translated to derive the generative mechanism through the retroduction based on the critical realism paradigm. Among the studies summarized in Table 9, Vaast et al. (2017) also take critical realism to develop a theory. Still, their focal point is the emerging roles of actors in social media and their interdependencies to enable collective actions. In their study, the unfolding processes, tweet threads, are modeled as interdependence among three types of users, ignoring the temporal order of actions taken by the users. Thus, this approach does not suit routine studies.

LIMITATIONS AND CONCLUSION

Our framework is not without limitations. The first limitation comes from the pitfall inherent in big data. Because information systems collecting the process data are not designed for researchers' needs, they collect all types of data, only some of which may be irrelevant to the researcher's focal problem. They also provide far too much data. Therefore, it is important to have clear data needs for the study and start with a parsimonious data ontology.

The second limitation is the reliability of the digital trace data. Howison et al. (2011) discussed several reliability and validity issues when using digital trace data for social network analysis. In particular, the system's reliability and practice and the reliability of system-generated data are the

most relevant issues for process studies. The system's reliability and practice concerns the system's use and whether users utilize it in the way for which it was designed. The use of the system may also change over time due to system updates or user changes. Understanding the actual use of the system helps researchers to get an accurate interpretation. The reliability of system-generated data mainly concerns the system's recording process. For example, time zone management and inconsistent recordings are two critical issues for process studies.

The third limitation is the reliability of the computer algorithm. A computer algorithm is commonly developed based on certain assumptions and constraints to achieve the best performance. It is imperative for researchers to fully understand those assumptions and constraints to obtain reliable results. For example, a matching penalty is a critical part of a multi-sequence alignment used to calculate the distance of two sequences. In natural science, a matching penalty is developed from prior studies and experiments. Given the variety of social problems, it is often hard to develop such a universal matching penalty. Thus, Wilson (2006) suggested using a low penalty where the actual penalty is unknown.

Lastly, as new computational techniques are developed rapidly in different disciplines, we may have overlooked other potential algorithms and models, especially in natural science. There are approaches to identify sequence motifs directly from the input sequence (D'haeseleer 2006), which essentially combine sequence mining and clustering in one model. For example, there are deterministic optimization-based approaches, such as the Mixture Model by Expectation-Maximization (Bailey and Elkan 1994), and probabilistic optimization-based approaches, such as "Gibbs sampling" (Thijs et al. 2002). Similarly, sequence mining techniques, such as concise sequence pattern mining, can discover concise representations of patterns aggregated from the original dataset (Fournier-Viger et al. 2017). However, we believe it is more important for social scientists to ensure the alignment between the algorithmic assumptions and the research

objectives than to adopt the latest techniques. By separating sequence mining and clustering, our approach offers a clearer view of how latent patterns are identified through a “trial-and-error” process of sequence mining. The adjustment of search needs to be supervised with theoretical considerations. It also provides better transparency to the researchers to follow the approximation process. Researchers would see how latent patterns identified from sequence mining can be transferred into primitive patterns through clustering. Such a capability is especially helpful when the choice of algorithms and their specifications may result in different outcomes. The importance of aligning the algorithmic assumptions and the research objectives is also why some of the state-of-the-art machine learning algorithms, such as deep learning, are also not appropriate for our framework, as the overall mining and clustering process is considered to be a black box that limits the interpretability (Chen et al. 2018; Lipton 2018).

Despite these limitations, we believe our computation framework for studying routines offers a useful alternative for scholars interested in building new process theories. We introduce sequence analytics as a methodological tool. Our framework brings a critical realist perspective with computational methods. It shows how sequential pattern mining and hierarchical cluster analysis can be combined to support process-oriented studies of routines by illustrating how to identify routines and their evolutions.

REFERENCES

Aaltonen, A., and Seiler, S. 2015. "Cumulative Growth in User-Generated Content Production: Evidence from Wikipedia," *Management Science* (62:7), pp. 2054-2069.

Abbott, A. 1990. "A Primer on Sequence Methods," *Organization Science* (1:4), pp. 375-392.

Abbott, A., and Barman, E. 1997. "Sequence Comparison Via Alignment and Gibbs Sampling: A Formal Analysis of the Emergence of the Modern Sociological Article," *Sociological Methodology* (27), pp. 47-87.

Abbott, A., and Forrest, J. 1986. "Optimal Matching Methods for Historical Sequences," *The Journal of Interdisciplinary History* (16), pp. 471-494.

Abbott, A., and Hrycak, A. 1990. "Measuring Resemblance in Sequence Data: An Optimal Matching Analysis of Musicians' Careers," *American Journal of Sociology* (96:1), pp. 144-185.

Abbott, A., and Tsay, A. 2000. "Sequence Analysis and Optimal Matching Methods in Sociology Review and Prospect," *Sociological Methods & Research* (29), pp. 3-33.

Agrawal, R. and Srikant, R., 1995, March. Mining sequential patterns. In *Proceedings of the eleventh international conference on data engineering (pp. 3-14). IEEE*.

Andrews, R.. Suriadi, S., Wynn, M., ter Hofstede, A.H.M., and Rothwell, S. 2018. “Improving Patient Flows at St. Andrew’s War Memorial Hospital’s Emergency Department Through Process Mining,” in J. vomBrocket, J. Mendling (eds.) *Business Process Management Cases*, Management for Professionals, DOI 10.1007/978-3-319-58307-5_17, pp. 311 – 333.

Archer, M., Decoteau, C., Gorski, P., Little, D., Porpora, D., Rutzou, T., Smith, C., Steinmetz, G., and Vandenberghe, F. 2016. "What Is Critical Realism?" Perspectives: A Newsletter of the ASA Theory Section, Fall 2017. Available at http://www.asatheory.org/current-newsletter-online/what-is-critical-realism

Bagozzi, R. P., and Dholakia, U. M. 2002. "Intentional Social Action in Virtual Communities," *Journal of Interactive Marketing* (16), pp. 2-21.

Bailey TL, Elkan C (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc. Int. Conf. Intell. Syst. Mol. Biol. 2, 28–36

Bargeman, B., Joh, C.-H., and Timmermans, H. 2002. "Vacation Behavior Using a Sequence Alignment Method," *Annals of Tourism Research* (29:2), pp. 320-337.

Becker, M.C., 2005. “The concept of routines: some clarifications,” *Cambridge Journal of Economics*, 29(2), pp.249-262.

Berente, N., Seidel, S., and Safadi, H. 2019. "Data-Driven Computationally-Intensive Theory Development," *Information Systems Research (30:1).*.

Bhaskar R. 1998. “Philosophy and scientific realism.” In: *Critical Realism: Essential Readings, Critical Realism: Interventions*, (eds) M. Archer, Routledge. pp. 16-47.

Bhaskar, R. 2013. *A Realist Theory of Science*. Routledge.

Bingham, CB. and Kahl, SJ. 2013) The process of schema emergence: Assimilation, deconstruction, unitization and the plurality of analogies. *Academy of Management Journal* 56(1):14–34.

Blair-Loy, M. 1999. "Career Patterns of Executive Women in Finance: An Optimal Matching Analysis," *American Journal of Sociology* (104:5), pp. 1346-1397.

Bygstad, B. 2010. "Generative Mechanisms for Innovation in Information Infrastructures," *Information and Organization* (20:3-4), pp. 156-168.

Chang, H. 2004. *Inventing Temperature: Measurement and Scientific Progress*. Oxford University Press.

Chang, R. M., Kauffman, R. J., and Kwon, Y. 2014. "Understanding the Paradigm Shift to Computational Social Science in the Presence of Big Data," *Decision Support Systems* (63), pp. 67-80.

Chen, N. C., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R. 2018. Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. ACM Transactions on Interactive Intelligent Systems (TiiS), 8(2), 1-20.

Cornwell, B. 2015. *Social Sequence Analysis: Methods and Applications*. Cambridge University Press.

Danermark, B., Ekstrom, M., and Jakobsen, L. 2001. *Explaining Society: An Introduction to Critical Realism in the Social Sciences*. Routledge.

DeLanda, M. 2013. *Intensive Science and Virtual Philosophy* (Bloomsbury Publishing).

Deleuze, G. 1994. *Difference and Repetition* (Columbia University Press).

D'haeseleer, P. 2006a. How does DNA sequence motif discovery work? *Nature Biotechnology* 24(8), pp. 959–961.

D'haeseleer, P. 2006b. What are DNA sequence motifs? *Nature Biotechnology* 24(4), pp. 423 - 425.

Dobson, P. J. 2001. "The Philosophy of Critical Realism—an Opportunity for Information Systems Research," *Information Systems Frontiers* (3:2), pp. 199-210.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2011. *Dynamic Structures of Control and Generativity in Digital Ecosystem Service Innovation: The Cases of the Apple and Google Mobile App Stores*. London School of Economics and Political Science.

Elder-Vass, D. 2010. *The Causal Power of Social Structures: Emergence, Structure and Agency*. Cambridge: Cambridge University Press.

Everaert, MB, Huybregts, MA, Chomsky, N, Berwick, RC, and Bolhuis, JJ. 2015. Structures, not strings: linguistics as part of the cognitive sciences. *Trends in Cognitive Sciences* 19(12) pp. 729–743.

Faraj, S., Jarvenpaa, S. L., and Majchrzak, A. 2011. "Knowledge Collaboration in Online Communities," *Organization science* (22), pp. 1224-1239.

Feldman, M.S. and Pentland, B.T. 2003. "Reconceptualizing Organizational Routines as a Source of Flexibility and Change", *Administrative Science Quarterly* (48:1), pp. 94 – 118.

Fleming, L., and Waguespack, D. M. 2007. "Brokerage, Boundary Spanning, and Leadership in Open Innovation Communities," *Organization Science* (18), pp. 165-180.

Fletcher, A. J. 2017. "Applying Critical Realism in Qualitative Research: Methodology Meets Method," *International Journal of Social Research Methodology* (20), pp. 181-194.

Fournier-Viger, P., Lin, J.C.W., Kiran, R.U., Koh, Y.S. and Thomas, R., 2017. "A survey of sequential pattern mining," *Data Science and Pattern Recognition,* 1(1), pp.54-77.

Gabadinho, A., Ritschard, G., Mueller, N. S., and Studer, M. 2011. "Analyzing and Visualizing State Sequences in R with Traminer," *Journal of Statistical Software* (40), pp. 1-37.

Garud, R. and Van De Ven, AH. 1992. An empirical evaluation of the internal corporate venturing process. *Strategic Management Journal* 13(S1):93–109.

Gaskin, J., Berente, N., Lyytinen, K., and Yoo, Y. 2014. "Toward Generalizable Sociomaterial Inquiry: A Computational Approach for Zooming in and out of Sociomaterial Routines," *MIS Quarterly* (38:3), pp. 849-871.

Gioia, D.A., Corley, K.G., Hamilton, A.L. 2013. "Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology," *Organizational Research Methods*, https://doi.org/10.1177/1094428112452151.

Giles, J. 2012. "Making the Links," *Nature* (488), p. 448.

Goh, KT, Pentland, BT. 2019. "From Actions to Paths to Patterning: Toward a Dynamic Theory of Patterning in Routines," *Academy of Management Journal* 62(6), pp. 1901–1929.

Grisold, T., Wurm, B., Mendling, J., and vom Brocke, J. 2020. "Using Process Mining to Support Theorizing About Change in Organizations," in Proceedings of the 53rd HICSS 2020, URI: https://hdl.handle.net/10125/64417 pp. 5492 – 5501.

Halfaker, A., Geiger, R. S., Morgan, J. T., and Riedl, J. 2012. "The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline," *American Behavioral Scientist*), p. 0002764212469365.

Halpin, B. 2010. "Optimal Matching Analysis and Life-Course Data: The Importance of Duration," *Sociological Methods & Research* (38:3), pp. 365-388.

Halpin, B., and Cban, T. W. 1998. "Class Careers as Sequences: An Optimal Matching Analysis of Work-Life Histories," *European Sociological Review* (14:2), pp. 111-130.

Hansson M, Pentland B, Hrem T. 2018. Identifying and Describing Characteristics of Organizational Routines as Repertoires of Action Patterns, Degree of Routinization, and Enacted Complexity. *Academy of Management Global Proceedings* (2018), pp. 37.

Henfridsson, O., and Bygstad, B. 2013. "The Generative Mechanisms of Digital Infrastructure Evolution.," *MIS Quarterly* (37:3), pp. 907 – 931.

Howard-Grenville, J., Rerup, C., Langley, A., & Tsoukas, H. 2016. "Introduction: Advancing a process perspective on routines by zooming out and zooming in." In J. Howard-Grenville et al., (eds.), *Organizational Routines: How they are Created, Maintained, and Changed. Perspectives on Process Organization Studies*, chapter 1: 1-18. Volume 6. Oxford: Oxford University Press.

Howison, J., Wiggins, A., and Crowston, K. 2011. "Validity Issues in the Use of Social Network Analysis with Digital Trace Data," *Journal of the Association for Information Systems* (12), p. 767.

Iriberri, A., and Leroy, G. 2009. "A Life-Cycle Perspective on Online Community Success," *ACM Computing Surveys (CSUR)* (41), p. 11.

Jungherr, A. and Theocharis, Y., 2017. "The empiricist's challenge: Asking meaningful questions in political science in the age of big data," *Journal of Information Technology & Politics* (14:2), pp. 97 - 109.

Ketchen Jr, D. J., and Shook, C. L. 1996. "The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique," *Strategic Management Journal*, pp. 441-458.

Kittur, A., and Kraut, R. E. 2008. "Harnessing the Wisdom of Crowds in Wikipedia: Quality through Coordination," *ACM*, pp. 37-46.

Klarner, P., and Raisch, S. 2013. "Move to the Beat—Rhythms of Change and Firm Performance," *Academy of Management Journal* (56), pp. 160-184.

Langley, A. 1999. “Strategies for Theorizing from Process Data,” *Academy of Management Review*, 24 (4), pp. 691 – 710.

Langley, A., Smallman, C., Tsoukas, H., and Van de Ven, A. H. 2013. "Process Studies of Change in Organization and Management: Unveiling Temporality, Activity, and Flow," *Academy of Management Journal* (56), pp. 1-13.

Langley A, Tsoukas H. 2016. *The SAGE handbook of process organization studies* (Sage).

Lawson, T. 1997. *Economics and Reality*. London: Routledge.

Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., and Gutmann, M. 2009. "Life in the Network: The Coming Age of Computational Social Science," *Science (New York, NY)* (323), p. 721.

Li X, Zhou F. 2017. “The Relationship Between Process Variability and Structural Connectivity in Open Source Software Development,” *ICIS 2017 Proceedings*.

Liben-Nowell, D., and Kleinberg, J. 2007. "The Link-Prediction Problem for Social Networks," *Journal of the Association for Information Science and Technology* (58), pp. 1019-1031.

Lindberg, A., Berente, N., Gaskin, J., and Lyytinen, K. 2016. "Coordinating Interdependencies in Online Communities: A Study of an Open Source Software Project," *Information Systems Research* (27:4), pp. 751-772.

Lipton, Z. C. 2018. “The mythos of model interpretability,” *Queue*, 16(3), 31-57.

MacIndoe, H. and Abbott, A., 2004. Sequence analysis and optimal matching techniques for social science data (pp. 387-406). in Handbook of Data Analysis.

Maguire, S. and Hardy, C. 2013. “Organizing processes and the construction of risk: A discursive approach,” *Academy of Management Journal* 56(1), pp. 231–255.

Mihaescu, R., Levy, D., and Pachter, L. 2009. "Why Neighbor-Joining Works," *Algorithmica* (54), pp. 1-24.

Miller, K. D. 2015. "Agent-Based Modeling and Organization Studies: A Critical Realist Perspective," *Organization Studies* (36), pp. 175-196.

Mingers, J. 2004. "Real-Izing Information Systems: Critical Realism as an Underpinning Philosophy for Information Systems," *Information and Organization* (14:2), pp. 87-103.

Mingers, J., Mutch, A., and Willcocks, L. 2013. "Critical Realism in Information Systems Research," *MIS Quarterly* (37:3), pp. 795-802.

Mooney, C. H., and Roddick, J. F. 2013. "Sequential Pattern Mining--Approaches and Algorithms," *ACM Computing Surveys (CSUR)* (45:2), p. 19.

Müller O, Junglas I, Brocke J vom, Debortoli S. 2016. “Utilizing big data analytics for information systems research: challenges, promises and guidelines,” *European Journal of Information Systems* 25(4), pp. 289–302.

Murray, F., and O'Mahony, S. 2007. "Exploring the Foundations of Cumulative Innovation: Implications for Organization Science," *Organization Science* (18), pp. 1006-1021.

O'Mahony, S., and Ferraro, F. 2007. "The Emergence of Governance in an Open Source Community," *Academy of Management Journal* (50), pp. 1079-1106.

Onnela, J.-P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., and Barabási, A.-L. 2007. "Structure and Tie Strengths in Mobile Communication Networks," *Proceedings of the National Academy of Sciences* (104), pp. 7332-7336.

Pei, J., Han, J., Mortazavi-Asl, B, Wang, J., Pinto, H., Chen, Q., Dayal, U, Hsu, M.C. 2004. “Mining sequential patterns by pattern-growth: the PrefixSpan approach,” *IEEE Transactions on Knowledge and Data Engineering* 16(11):1424–1440.

Pentland, B. 2003. "Conceptualizing and Measuring Variety in the Execution of Organizational Work Processes," *Management Science* (49:7), pp. 857-870.

Pentland, B. T. 1999. "Building Process Theory with Narrative: From Description to Explanation," *Academy of Management Review* (24), pp. 711-724.

Pentland, B. T., and Feldman, M. S. 2005. "Organizational Routines as a Unit of Analysis," *Industrial and Corporate Change* (14), pp. 793-815.

Pentland, B.T., Feldman, M.S., Becker, M.C., and Liu, P. 2012. "Dynamics of Organizational Routines: A Generative Model," *Journal of Management Studies* (49:8), pp. 1484 – 1508.

Pentland, B. T., Hærem, T., and Hillison, D. 2010. "Comparing Organizational Routines as Recurrent Patterns of Action," *Organizational Studies* (31:7), pp. 917-940.

Pentland, B. T., and Hærem, T. 2015. "Organizational Routines as Patterns of Action: Implications for Organizational Behavior," *Annual Review of Organizational Psychology and Organizational Behavior* (2), pp. 465 – 487.

Pentland BT, Rueter HH. 1994. "Organizational Routines as Grammars of Action," *Administrative Science Quarterly* 39(3):484–510.

Pilny A, Schecter A, Poole MS, Contractor N. 2016. "An illustration of the relational event model to analyze group interaction processes," *Group Dynamics: Theory, Research, and Practice* 20(3):181.

Ransbotham, S., and Kane, G. C. 2011. "Membership Turnover and Collaboration Success in Online Communities: Explaining Rises and Falls from Grace in Wikipedia," *MIS Quarterly-Management Information Systems* (35), p. 613.

Reddy, C.K. and Vinzamuri, B., 2013. A Survey of Partitional and Hierarchical Clustering Algorithms. In Charu C Aggarwal and Chandan K. Reddy (Eds.), Data clustering: Algorithms and Applications (pp. 87 – 110). CRC Press, Taylor & Francis Group.

Ritschard, G., Bürgin, R., and Studer, M. 2013. "Exploratory Mining of Life Event Histories," In J.J. McArdle & G. Ritschard (eds), *Contemporary Issues. in Exploratory Data Mining in Behavioral Sciences*, pp. 221-253. Routeledge, New York.

Sabherwal R and Robey D. 1993. "An empirical taxonomy of implementation processes based on sequences of events in information system development," *Organization Science* 4(4):548–576.

Salvato, C. and Rerup, C. 2011. "Beyond Collective Entities: Multilevel Research on Organizational Routines and Capabilities," *Journal of Management* (37:2), pp. 468 – 490.

Schecter, A., Pilny, A., Leung, A., Poole, M. S., and Contractor, N. 2018. "Step by step: Capturing the dynamics of work team process through relational event sequences," *Journal of Organizational Behavior* 39 (9), pp.1-19.

Shoval, N., and Isaacson, M. 2007. "Sequence Alignment as a Method for Human Activity Analysis in Space and Time," *Annals of the Association of American Geographers* (97), pp. 282-297.

Shoval, N., McKercher, B., Birenboim, A., and Ng, E. 2015. "The Application of a Sequence Alignment Method to the Creation of Typologies of Tourist Activity in Time and Space," *Environment and Planning B: Planning and Design* (42:1), pp. 76-94.

Smith, M. L. 2006. "Overcoming Theory-Practice Inconsistencies: Critical Realism and Information Systems Research," *Information and Organization* (16:3), pp. 191-211.

Stovel, K., and Bolan, M. 2004. "Residential Trajectories: Using Optimal Alignment to Reveal the Structure of Residential Mobility," *Sociological Methods & Research* (32:4), pp. 559-598.

Studer, M., and Ritschard, G. 2014. "A Comparative Review of Sequence Dissimilarity Measures," Working Paper, DOI : 10.12682/lives. 2296-1658.2014.33, available at *https://archive-ouverte.unige.ch/unige:78575*.

Thijs, G, Marchal, K, Lescot, M, Rombauts, S, De Moor, B, Rouzé, P, and Moreau, Y. 2002. "A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes," *J. Comput. Biol.* 9(2), pp. 447–464.

Vaast, E, Safadi, H, Lapointe, L, and Negoita, B. 2017. "Social Media Affordances for Connective Action: An Examination of Microblogging Use During the Gulf of Mexico Oil Spill," *MIS Quarterly* 41(4).

Van de Ven, A. H. and Poole, M. S. 1995. "Explaining Development and Change in Organizations," *Academy of Management Review* (20), pp. 510-540.

Van der Aalst, W.M.P.. 2011. *Process mining – Discovery, conformance and Enhancement of Business Processes*, Springer, Berlin, Heidelberg, DOI https://doi.org/10.1007/978-3-642-19345-3.

Van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., Van Dongen, B.F., Alves de Medeiros, A.K., Song, M., and Verbeek, H.M.W. 2007. "Business process mining: An industrial application," *Information Systems* (32), pp. 713 – 732.

Volkoff, O., Strong, DM., and Elmes, MB. 2007. "Technological embeddedness and organizational change," *Organization Science* 18(5):832–848.

Williams, C. K., and Karahanna, E. 2013. "Causal Explanation in the Coordinating Process: A Critical Realist Case Study of Federated It Governance Structures," *MIS Quarterly* (37:3), pp. 933-964.

Wilson, C. 2001. "Activity Patterns of Canadian Women: Application of Clustalg Sequence Alignment Software," *Transportation Research Record: Journal of the Transportation Research Board*, pp. 55-67.

Wilson, C. 2006. "Reliability of Sequence-Alignment Analysis of Social Processes: Monte Carlo Tests of Clustalg Software," *Environment and Planning A* (38), pp. 187-204.

Wynn, D., and Williams, C. K. 2012. "Principles for Conducting Critical Realist Case Study Research in Information Systems," *MIS Quarterly* (36), pp. 787-810.

Zachariadis, M., Scott, S. V., and Barrett, M. I. 2013. "Methodological Implications of Critical Realism for Mixed-Methods Research.," *MIS Quarterly* (37), pp. 855-879.

Zhang, Z., Choi, Y., Lee, H., and Yoo, Y. 2016. "The Organizing of Online Community: Fluid and Structural Organizing," In *Proceedings of ICIS2016*, Dublin, Ireland, 10 - 14 Dec 2016.

APPENDIX A. Summary of Sequential Pattern Mining Techniques

A pervasive assumption underlying sequence analysis is that elements do not appear at random, and, accordingly, a fundamental goal of sequence analysis is to detect patterns in the element order. There is usually a stochastic pattern underlying the sequence structure in target sequence data, meaning that certain elements tend to precede and follow certain others (Cornwell 2015). Likewise, in many event-sequence cases, the order in which events tend to appear within a sequence leading up to a given position helps one to predict which event will appear in that position. Most sequence analysis techniques start with detecting hidden stochastic patterns. This is why most techniques have their mathematical origin from first-order Markov models, the semi-Markov process, Chapman-Kolmogorov equations, and so on. The process of identifying patterns from sequence data is a very significant step to develop an approximation model that is closer to reality.

In general, there are three main types of sequential pattern mining algorithms: breadth-first search algorithms, depth-first search algorithms, and pattern-growth algorithms, the latter being another type of depth-first algorithm which is designed to address a limitation of depth-first search (Fournier-Viger et al. 2017). All these types are regarding how to find a frequent sequential pattern and are more frequently based on the pervasive assumption mentioned above.

Breadth-first search algorithms scan the data to find frequent 1-sequences (sequential patterns containing a single item). Then, they generate 2-sequences by adding another item as a prefix or suffix (so-called s-extensions and i-extensions). In this way, they increase the size of the sequence until no sequences can be generated. AprioriAll, which was first proposed by Agrawal and Srikant (1995), follows the same way to search for the frequent sequence. However, fundamental breadth-first algorithms, such as AprioriAll, can easily become inefficient depending

on the amount of data and the number of elements, because it causes a large number of subsets. To improve this inefficiency, many studies have proposed adjusted algorithms even though their methodological origin lies in the breadth-first search (GSP, PSP, SPIRIT, and MFS, etc.). Table A1 shows a comprehensive summary of representative breadth-first search algorithms for sequence mining.

Table A1. Common breadth-first sequence mining algorithms

Algorithms	Reference	Note
AprioriAll	Agrawal and Sirkant 1995	Original Apriori algorithm
Generalized Sequential Patterns (GSP)	Srikant and Agrawal 1996	Applies time constraints (the period between adjacent elements in a pattern).
PSP	Masseglia et al. 1998	Extend GSP algorithm by organizing candidate sequences in a prefix-tree.
Sequential Pattern Mining with Regular expression consTraints (SPIRIT)	Garofalakis et al. 1999	Use regular expressions as a flexible constraints specification tool
Maximum Frequent Sequence (MFS)	Zhang et al. 2001	Improves GSP

Depth-first search algorithms, such as Spade, Spam, Lapin, CM-SPAM, and CM-Spade, explore the search space of patterns by following different orders from the breadth-first search. They start from the sequences containing all possible single items and then recursively perform the extension of sequences to generate large sequences. Then, when a pattern cannot be extended longer, the algorithm backtracks to generate possible patterns using other sequences. As an important sub-branch of depth-first algorithms, pattern-growth algorithms (PrefixSpan, FreeSpan) are designed to address a limitation of the depth-first search, which is to generate candidate patterns that may not be found in the data. Pattern-growth algorithms can avoid this problem by recursively scanning the data to find larger patterns. Thus, they consider only the patterns actually

appearing in the database. Table A2 shows a comprehensive summary of representative depth-first algorithms and pattern-growth algorithms for sequence mining.

Table A2. Common depth-first search and pattern-growth sequence mining algorithms

Algorithms	Reference	Note
Spade	Zaki 2001	Utilizes vertical database representation for efficient depth-first search
Sequential Pattern Mining (SPAM)	Ayres et al. 2002	Applies depth-first search strategy
Last Position Induction(LAPIN)	Yang et al. 2007	Reduces search space
CM-SPAM / CM-SPADE	Fournier-Viger et al. 2014	Based on co-occurrence pruning
PREFIX-projected Sequential Pattern Mining (PrefixSpan)	Pei et al. 2004	*Pattern-growth (Applies recursive dataset projection to partition the data for efficiency)
Frequent Pattern-projected Sequential Pattern mining (FreeSpan)	Han et al. 2000	*Pattern-growth

The choice of specific algorithms is thus more constrained by the tools the researcher is comfortable with, especially for researchers without strong computer-science training. Table A3 shows the currently available popular software tools for sub-sequence mining.

Table A3. Examples of available software for sub-sequence mining.

Programming Language	Library/Package	Algorithm
R	TraMineR[20]	Apriori-based prefix-tree
R	arulesSequences[21]	Apriori-based cSPADE
Java	SPMF[22]	Supports a wide range of algorithms including both approaches.
Spark	MLlib[23]	Pattern-growth based PFP and PrefixSpan

[20] http://traminer.unige.ch
[21] https://cran.r-project.org/web/packages/arulesSequences/index.html
[22] http://www.philippe-fournier-viger.com/spmf/
[23] http://spark.apache.org/docs/2.2.0/mllib-frequent-pattern-mining.html

Python	pymining[24]	Pattern-growth based algorithms
Python	PyFIM[25]	Supports a wide range of algorithms including both approaches.
SPMF		Offers implementations of more than 170 data mining algorithms for discovering sequential patterns, sequential rules, association rules, high utility patterns, frequents item-sets, periodic patterns, clusters, and more (Java open-source library).

Aside from minimal support and max length, there are few important criteria, such as the gap and counting method. Figure A1 shows an illustration result based on three sequences. Two sub-sequences can be identified: (A-B-B) in the solid-line box and (D-B-A-C-B) in the dash-line box.

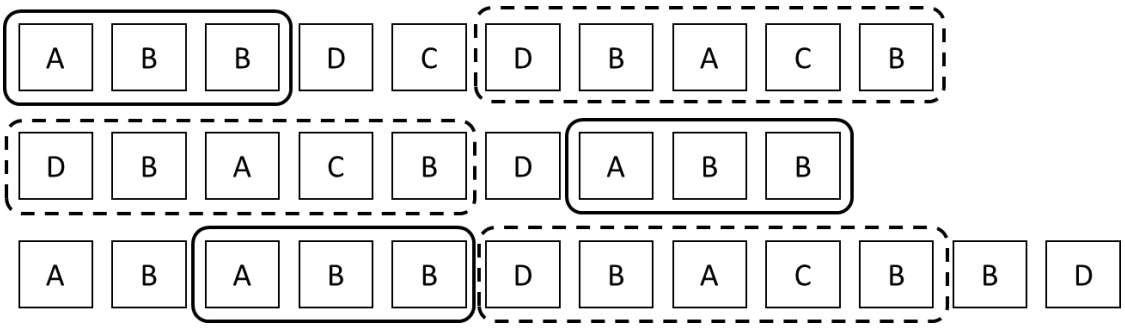

Figure A1. Illustration of sequential pattern mining result.

The maximum gap specifies the number of gap events allowed in a sub-sequence. For example, if one gap is allowed, then (A-B-C-B) will be seen as being the same as (A-B-A-B). A gap is usually helpful when the potential sub-sequence is sufficiently long that a minor difference can be ignored. For example, in searching for motifs, biologists will normally allow gaps.

The counting method specifies how the occurrence of a sub-sequence is counted. Generally, it can be as follows: 1) count once for each occurrence with no overlap allowed between two sub-

[24] https://github.com/bartdag/pymining
[25] http://www.borgelt.net/pyfim.html

sequences; 2) count once for each occurrence with overlap allowed between two sub-sequences; or 3) count once as long as the occurrence is observed, regardless of the number of occurrences. In the illustrative section and appendix B, we discuss in more detail the choices of key parameters.

APPENDIX B. Summary of key parameters for sub-sequence mining in TraMineR

Table B1. The summary of key parameters for sub-sequence mining in TraMineR

Parameters	Explanation	Consideration
min.support	It specifies the threshold (in number of occurrences) that a sub-sequence needs to meet to be classified as a frequent sub-sequence	High min.support will allow researchers to identify these highly frequent sub-sequences, while low min.support will allow researchers to explore a greater number of less frequent sub-sequences at the cost of longer running time. We recommend starting with low support (with a smaller sample if the full dataset is too large). First, it helps researchers to gain a better idea about the data and find the more appropriate support number; second, in social science, any frequency greater than pure random chance could mean a potentially interesting pattern. We used pmin.support as the preferred parameter in this case.
pmin.support	It specifies the threshold (in percentage of occurrences) that a sub-sequence needs to meet to be classified as a frequent sub-sequence	Similar to min.support, pmin.support uses percentage instead of count as the threshold. Generally, we recommend using percentage over number count to determine the frequent sub-sequence. We initially started with 20% but found too many matching sub-sequences and ended up with 30% to capture more potential interesting sub-sequences while kept the number manageable.
max.k	It specifies the maximum length that a sub-sequence can be	A larger number of max.k allows us to find longer potential sub-sequences, at the cost of longer running time. It is not always necessary to set max.k as large as possible. First, it depends on the research context, whether the long sub-sequence is possible, interesting and realistic; second, longer sub-sequences are often just repetitions of short sub-sequences. Again, this needs to be determined through the iterative tests of different settings. We recommend starting with a larger number (with a smaller number if the full dataset is too large). We initially started with 10 but found too many matching sub-sequences with lots of repetitions and ended up with 8 based on the two points discussed above.
Constraint	It specifies the gap constraint and counting methods for searching for a sub-sequence, which includes a set of its own parameters. The key parameters are shown below.	

max.gap	It specifies the maximum gap allowed between two events in a sub-sequence. No gap by default.	The consideration here is similar to the choice of min.support. The use of gap may allow us to find otherwise ignored sub-sequences that contain few noises. It will also increase the running time. It can be used as a way to explore the data at the beginning. The decision should also be mainly theory driven, querying whether similar sub-sequences with minor gap should be considered to be the same. We set max gap of 1 in our case: given the maximum length of 8, one gap means around 12% difference, and we consider two sequences with more than one gap to be two distinct sequences.
window.size	the moving window size for CWIN and CMINWIN counting methods.	This is related to the counting method discussed below. It should be decided based on the combination of max.gap and max.k. We did not use the window-based counting method in this case.
count.method	It specifies the method used to count the occurrences of a sub-sequence. “COBJ” is the default method and only counts whether the sub-sequence occurs in the sequence. “CWIN” differs from COBJ by counting whether the sub-sequence occurs in the window rather than the whole sequence. “CMINWIN” differs from CWIN by counts the number of minimal windows containing the occurrence. “CDIST_O” counts the number of distinct occurrences with the possibilities of overlap between two occurrences. “CDIST” counts the number of distinct occurrences without the possibility of overlap between two occurrences.	As we can see from the description, the choice of counting method depends on the goal and context of the research. Joshi and his colleagues (1999) have a detailed discussion on the differences between and the implications of the methods. In short, COBJ is concerned only with whether an occurrence of sub-sequence exists in the sequence, which is less meaningful in searching for common patterns, but useful for examining how commonly the sub-sequence may be observed in different sequences. CDIST_O and CWIN are similar in the sense that they both count the number of all occurrences (or windows containing the occurrence). CDIST and CMINWIN are similar in the sense that they both count the number of minimal occurrences (or windows containing the occurrence). Whether to allow overlap between occurrences (or windows) is something for the researcher to experiment with. If there are only a few types of events (4 in our case) and the main interest is on shorter sub-sequence, then the chance of overlapping will be very high (for example, AAAA will be counted 4 times in AAAAAAA), in this case, CDIST will be preferred to CDIST_O to minimize the overestimation. We chose CDIST for the same reason.

APPENDIX C. Sequential Pattern Mining Results

We choose the most frequent sub-sequence from each cluster as the representative sub-sequence: three continuous non-creator editings ((NCED)-(NCED)-(NCED)) for non-creator dominated sequences, three continuous creator editings ((CRED)-(CRED)-(CRED)) for creator-dominated short sequences, and eight continuous creator editings (CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)) for creator-dominated long sequences. These sequences can give us a sense of how creators and other non-creator editors work iteratively. Continuous editing shows the degree of domination from the creator. Table C1 shows the occurrence of each representative sub-sequence at different periods. For example, we can find (NCED)-(NCED)-(NCED) in 36.8% featured articles created between 2001 and the end of 2003.

Table C1. Changes of occurrence of three representing editing patterns over time[26].

Representing Patterns	Article	2001-2003	2004-2006	2007-2009	2010-2012	2013-2015
(NCED)-(NCED)-(NCED)	Featured	36.8%	62.1%*	71.1%*	67.5%	72.5%
	Non-featured	<30%	39.1%*	33.3%*	34.6%	34.4%
(CRED)-(CRED)-(CRED)	Featured	<30%	33.9%*	58.5%*	67.5%*	71.9%
	Non-featured	<30%	31.0%	33.0%	37.9%	39.7%
(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	Featured	<30%	<30%	39.1%*	49.5%*	55.6%
	Non-featured	<30%	<30%	<30%	<30%	<30%

[26] Star sign indicates the change is statistically significant from the previous period.

For articles created from 2001 to 2003, we show in Table C2 all sub-sequences that have occurred in at least 20% of articles. This is to help readers to understand the analysis results. From the table below, we can see that the collaboration patterns do not differ significantly from each other, with (CRED)-(NCED)-(NCED) being the most frequent "patterns" for both featured articles and non-featured articles.

Table C2. Frequent Sub-Sequences for Wikipedia Articles Created between 2001 and 2003

Featured Article			Non-featured Article		
	Sub-Sequence	Percentage		Sub-Sequence	Percentage
2	(CRED)-(NCED)-(NCED)	38.45%	2	(CRED)-(NCED)-(NCED)	28.38%
1	(NCED)-(NCED)-(NCED)	36.80%	2	(CRED)-(CRED)-(NCED)	22.56%
1	(NCED)-(NCED)-(NCED)-(NCED)	27.10%			
1	(CRED)-(NCED)-(NCED)-(NCED)	26.03%			
2	(CRED)-(CRED)-(NCED)	20.54%			
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	20.19%			

For featured articles (Table C3-1) created during 2004 and 2006, the types of repeated collaboration patterns did not change much from those of articles created during the previous time period, but the percentage of articles with similar collaboration patterns greatly increased. For example, the continuous non-creator edits (NCED)-(NCED)-(NCED) increased from 36% to 62%. Similar increases can also be found for other patterns identified in the previous period. Although the types of sub-sequence are largely the same, there is a new one (CRED)-(CRED)-(CRED) appearing in 34% articles, which shows the emergence of creator-dominated patterns.

Table C3-1. Frequent Sub-Sequences for Featured Articles Created between 2004 and 2006

Featured Article

Cluster	Sub-Sequence	Percentage
1	(NCED)-(NCED)-(NCED)	62.07%
2	(CRED)-(NCED)-(NCED)	57.05%
1	(NCED)-(NCED)-(NCED)-(NCED)	50.48%
1	(CRED)-(NCED)-(NCED)-(NCED)	44.49%
2	(CRED)-(CRED)-(NCED)	44.17%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	42.31%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	35.16%
1	(CRED)-(NCED)-(NCED)-(NCED)-(NCED)	34.19%
2	(CRED)-(CRED)-(CRED)	33.93%
2	(CRED)-(CRED)-(NCED)-(NCED)	33.10%

For non-featured articles created during the same period (Table C3-2), a similar change also applies with two new non-creator-dominated patterns. All four identified sub-sequences can also be found in featured articles with similar percentages. This suggests that until now, the collaboration patterns between the two types of articles did not show a big difference. The increased number of sub-sequence types for featured articles is more likely a result of the increased total number of edits. Compared with featured articles in the same period, we did not see the creator-dominated patterns like (CRED)-(CRED)-(CRED).

Table C3-2. Frequent Sub-Sequences for Non-featured Articles Created between 2004 and 2006

Non-featured Article		
Cluster	Sub-Sequence	Percentage
2	(CRED)-(NCED)-(NCED)	52.20%
2	(CRED)-(CRED)-(NCED)	40.84%
1	(NCED)-(NCED)-(NCED)	39.08%
1	(CRED)-(NCED)-(NCED)-(NCED)	36.33%

Since the year 2007, the collaboration patterns for featured articles have changed dramatically with many more new types of sub-sequences (as shown in Table C4-1). The increased number of sub-sequence types mostly arise from the increased length of the identified sub-sequences, since we can now see far longer sub-sequences. Given the dramatically increased total edits (from an average of 17 in the last period to an average of 47), it is not surprising to see such changes. However, despite the changes in the increased sequence length, we also see a rise in creator dominations in editing, as several sub-sequences with solely creator edits became popular, such as (CRED)-(CRED)-(CRED)-(CRED).

Table C4-1. Frequent Sub-Sequences for Featured Articles Created between 2007 and 2009

Featured Article		
Cluster	Sub-Sequence	Percentage
1	(NCED)-(NCED)-(NCED)	71.09%
1	(NCED)-(NCED)-(NCED)-(NCED)	61.45%
2	(CRED)-(CRED)-(CRED)	58.52%
2	(CRED)-(CRED)-(NCED)	55.17%
1	(NCTK)-(NCED)-(NCED)	54.75%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	53.77%
3	(CRED)-(CRED)-(CRED)-(CRED)	53.35%
2	(CRED)-(NCED)-(NCED)	51.12%
2	(NCED)-(NCED)-(NCTK)	50.70%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	49.02%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	46.79%
2	(CRED)-(CRED)-(CRED)-(NCED)	46.79%
1	(NCTK)-(NCED)-(NCED)-(NCED)	46.23%
2	(NCED)-(NCTK)-(NCED)	45.81%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	45.67%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	43.16%

3	(NCED)-(CRED)-(CRED)	42.18%
1	(NCED)-(NCED)-(NCED)-(NCTK)	41.76%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	41.62%
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	40.50%
1	(NCED)-(NCED)-(CRED)	40.22%
1	(CRED)-(NCED)-(NCED)-(NCED)	39.94%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)	39.25%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	39.11%
2	(CRED)-(CRED)-(NCED)-(NCED)	38.69%
1	(NCED)-(NCTK)-(NCED)-(NCED)	38.27%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	37.85%
3	(NCED)-(CRED)-(CRED)-(CRED)	36.45%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)	35.89%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	35.75%
2	(NCED)-(NCED)-(NCTK)-(NCED)	34.64%
2	(NCED)-(NCTK)-(NCTK)	34.36%
2	(CRED)-(CRED)-(CRTK)	33.94%
2	(CRED)-(CRED)-(NCTK)	33.52%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	33.38%
2	(CRED)-(NCED)-(CRED)	33.10%

Meanwhile, for non-featured articles (Table C4-2), although the average edits also slightly increased from 8 to 8.8 during the same period, we actually saw a decrease in terms of types as well as the percentage of sub-sequences. It prompts the question of whether the changes in the collaboration patterns are solely due to the changes of total edits.

Table C4-2. Frequent Sub-Sequences for Non-featured Articles Created between 2007 and 2009

Non-featured Article

Cluster	Sub-Sequence	Percentage
2	(CRED)-(NCED)-(NCED)	39.64%
2	(CRED)-(CRED)-(NCED)	35.38%
1	(NCED)-(NCED)-(NCED)	33.28%

Compared with the previous period, we start to see some interesting changes for articles created during 2010 and 2012 in Table C5-1. The first one is the increasing percentage of articles with continuous creator edits. Such patterns, like (CRED)-(CRED)-(CRED), have risen to be the most common patterns, with about a 10% increase from the previous period, being found in more than two-thirds of featured articles. This change however cannot be found for those non-creator-dominated patterns, such as (NCED)-(NCED)-(NCED). In fact, there is actually a slight decrease in those patterns. Another change is the rise in conversations among contributors, as we now see sub-sequences with talk only - 36% articles have (NCTK)-(NCTK)-(NCTK), which means contributors started to communicate before an edit was made.

Table C5-1. Frequent Sub-Sequences for Featured Articles Created between 2010 and 2012

Featured Article		
Cluster	Sub-Sequence	Percentage
2	(CRED)-(CRED)-(CRED)	67.47%
1	(NCED)-(NCED)-(NCED)	67.47%
3	(CRED)-(CRED)-(CRED)-(CRED)	62.63%
1	(NCED)-(NCED)-(NCED)-(NCED)	59.17%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	58.13%
2	(CRED)-(CRED)-(NCED)	58.13%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	55.71%
1	(NCTK)-(NCED)-(NCED)	55.36%
2	(NCED)-(NCED)-(NCTK)	52.60%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	52.25%

1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	51.21%
2	(CRED)-(CRED)-(CRED)-(NCED)	50.52%
2	(CRED)-(NCED)-(NCED)	50.17%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	49.83%
2	(NCED)-(NCTK)-(NCED)	48.79%
3	(NCED)-(CRED)-(CRED)	48.44%
2	(CRED)-(CRED)-(CRTK)	47.40%
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	44.98%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	44.29%
1	(NCTK)-(NCED)-(NCED)-(NCED)	44.29%
3	(NCED)-(CRED)-(CRED)-(CRED)	43.94%
1	(NCED)-(NCED)-(CRED)	43.94%
2	(CRED)-(CRED)-(CRED)-(CRTK)	43.94%
2	(CRED)-(NCED)-(CRED)	42.56%
1	(NCED)-(NCED)-(NCED)-(NCTK)	41.87%
2	(CRED)-(CRTK)-(CRED)	41.87%
3	(CRTK)-(CRED)-(CRED)	40.48%
2	(CRED)-(CRED)-(NCTK)	40.48%
2	(CRED)-(CRED)-(NCED)-(NCED)	40.48%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	40.14%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	40.14%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	39.10%
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)	38.75%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)	38.06%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	38.06%
1	(CRED)-(NCED)-(NCED)-(NCED)	37.72%
1	(NCED)-(NCTK)-(NCED)-(NCED)	37.37%
2	(NCTK)-(NCTK)-(NCTK)	36.68%
2	(NCED)-(NCED)-(NCTK)-(NCED)	36.68%

3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	36.68%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	36.33%
2	(NCED)-(NCTK)-(NCTK)	36.33%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)	35.99%
1	(NCED)-(NCED)-(CRED)-(CRED)	35.29%
3	(CRTK)-(CRED)-(CRED)-(CRED)	35.29%
2	(NCTK)-(CRED)-(CRED)	34.95%
2	(CRED)-(CRED)-(NCED)-(CRED)	34.95%
2	(NCTK)-(NCTK)-(NCED)	34.60%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	34.60%
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	34.26%
2	(CRED)-(CRED)-(CRED)-(NCTK)	34.26%
3	(CRED)-(CRTK)-(CRED)-(CRED)	33.56%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	33.22%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	33.22%

However, for non-featured articles created during the same period (Table C5-2), the collaboration patterns remain basically the same.

Table C5-2. Frequent Sub-Sequences for Non-featured Articles Created between 2010 and 2012

Non-featured Article		
Cluster	Sub-Sequence	Percentage
2	(CRED)-(NCED)-(NCED)	39.00%
2	(CRED)-(CRED)-(NCED)	38.13%
2	(CRED)-(CRED)-(CRED)	37.88%
1	(NCED)-(NCED)-(NCED)	34.63%

For featured articles created between 2013 and 2015 (Table C6-1), the sub-sequence types remain mostly the same as in previous years while the percentage continuously increases.

Table C6-1. Frequent Sub-Sequences for Featured Articles Created between 2013 and 2015

Featured Article		
Cluster	Sub-Sequence	Percentage
1	(NCED)-(NCED)-(NCED)	72.51%
2	(CRED)-(CRED)-(CRED)	71.93%
3	(CRED)-(CRED)-(CRED)-(CRED)	69.59%
1	(NCED)-(NCED)-(NCED)-(NCED)	66.08%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	65.50%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	61.99%
2	(CRED)-(CRED)-(NCED)	61.40%
2	(NCED)-(NCED)-(NCTK)	60.82%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	60.23%
3	(NCED)-(CRED)-(CRED)	59.06%
1	(NCTK)-(NCED)-(NCED)	58.48%
2	(CRED)-(CRED)-(CRED)-(NCED)	57.31%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	55.56%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	54.97%
2	(NCED)-(NCTK)-(NCED)	54.97%
2	(CRED)-(CRED)-(CRTK)	54.97%
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	53.80%
1	(NCED)-(NCED)-(NCED)-(NCTK)	53.22%
3	(NCED)-(CRED)-(CRED)-(CRED)	52.05%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	50.88%
2	(CRED)-(CRED)-(CRED)-(CRTK)	49.71%
1	(NCTK)-(NCED)-(NCED)-(NCED)	47.95%
2	(CRED)-(NCED)-(NCED)	47.95%
1	(NCED)-(NCED)-(CRED)	47.37%
2	(CRED)-(CRED)-(NCTK)	47.37%
3	(CRTK)-(CRED)-(CRED)	47.37%

2	(CRED)-(NCED)-(CRED)	46.78%
1	(NCED)-(NCTK)-(NCED)-(NCED)	46.20%
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)	46.20%
2	(CRED)-(CRTK)-(CRED)	45.61%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)	45.03%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	45.03%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	45.03%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	44.44%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)	43.27%
2	(NCED)-(NCED)-(NCTK)-(NCED)	43.27%
2	(CRED)-(CRED)-(NCED)-(NCED)	42.69%
2	(NCED)-(NCTK)-(NCTK)	41.52%
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	41.52%
2	(NCTK)-(CRED)-(CRED)	41.52%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	40.94%
1	(NCED)-(NCED)-(CRED)-(CRED)	40.94%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	40.94%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)	39.77%
2	(CRED)-(CRED)-(NCED)-(CRED)	39.77%
3	(CRED)-(NCED)-(CRED)-(CRED)	39.18%
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)	38.60%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	38.60%
2	(CRED)-(CRED)-(CRED)-(NCTK)	38.60%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	38.01%
3	(CRED)-(CRTK)-(CRED)-(CRED)	38.01%
3	(CRTK)-(CRED)-(CRED)-(CRED)	37.43%
2	(CRED)-(CRED)-(CRTK)-(CRED)	37.43%
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)	36.84%
1	(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)	36.26%

3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)	36.26%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	36.26%
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)	35.67%
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)	35.67%
2	(CRED)-(CRED)-(CRED)-(NCED)-(CRED)	35.67%
2	(NCTK)-(NCED)-(NCTK)	35.67%
2	(CRED)-(CRED)-(CRED)-(NCED)-(NCED)	35.67%
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)	35.09%
1	(CRED)-(NCED)-(NCED)-(NCED)	35.09%
2	(NCTK)-(CRED)-(CRED)-(CRED)	35.09%
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)	34.50%
3	(CRED)-(CRED)-(NCED)-(CRED)-(CRED)	34.50%
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)	34.50%
2	(NCTK)-(NCTK)-(NCTK)	33.92%
3	(CRED)-(NCED)-(CRED)-(CRED)-(CRED)	33.92%
3	(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)	33.92%

For non-featured articles (Table C6-2), the collaboration patterns remain basically the same.

Table C6-2. Frequent Sub-Sequences for Non-featured Articles Created between 2013 and 2015

Non-featured Article		
Cluster	Sub-Sequence	Percentage
2	(CRED)-(CRED)-(CRED)	39.73%
2	(CRED)-(CRED)-(NCED)	39.58%
2	(CRED)-(NCED)-(NCED)	38.69%
1	(NCED)-(NCED)-(NCED)	34.38%

APPENDIX D. Summary of Sequence Dissimilarity Measures

Measuring dissimilarity between sequences is the most common and significant starting point for sequence clustering. The dissimilarity measure can be formulated through a mathematical definition of distance between sequences. There are three main approaches to calculating the distance between sequences: distances between probability distribution; distance based on counts of common attributes; and optimal matching (Studer and Ritschard 2014).

To measure the distance between probability distribution, all sequences should be mapped into the probability vector based on the distribution of each element in the sequence. Then the Euclidean or Chi-square distance between the vectors can be calculated. The Euclidean distance accounts for the absolute differences in the proportion of time spent in the states. The squared Chi-square distance weights the squared differences for each state by the inverse of the overall proportion of time spent in the state, which, for two identical differences, gives more importance to a rare state than to a frequent state (Deville and Saporta 1983).

The idea of distances based on counts of common attributes has been prompted by error detection techniques for codes and writing, which was the main research area of early computational linguistics (Hamming 1950). The simplest measure can be a position-wise comparison between two given sequences and called Hamming distance (Hamming 1950). Measuring the length of the longest common prefix, suffix, and subsequence (Elzinga 2006) can be another option and is not sensitive to timing, unlike the position-wise comparison. The sequence vector representation-based matric (SVR) originated from the same idea but showed an evolved performance, as SVR allows us to weight each matched subsequence by its length (Elzinga and Studer 2015).

Optimal matching (OM) is the most common way of computing dissimilarities between sequences in the social science field (Studer and Ritschard 2016). The method borrows from other fields that use similar edit approaches, such as the Levenshtein distance in computer science (Levenshtein 1966; Li and Liu 2007) and sequence alignment in bioinformatics. OM (Sankoff and Kruskal 1983) measures the distance between two sequences as the minimum total cost of transforming one sequence into the other sequence through indels (insert or delete operations). Each operation is assigned a cost, which may vary with the involved states (Studer and Ritschard 2014). A key issue with OM is the choice of cost. Setting the proper cost can be achieved in different ways. One way is to use theory-based cost, which is applicable when prior studies can provide well-supported evidence of the possible cost for transformation from one event to another event. Another one is data-driven cost, where the substitution rate can be calculated from previous studies. Sequencing in biology is generally based on previous data.

Representative distance measures for each category are summarized in Table D1 below. The choice of distance measure must reflect the nature of the problem. If scholars are interested in explaining changes in timing, we need measures sensitive to timing. Position-wise measures, such as those of the Hamming family, are the most time-sensitive. Using the CHI2 and EUCLID distances, with the number of periods K equal to the sequence length, is also a solution (Studer and Ritschard 2016). If the focus is on changes in the sequencing (the order of events in the sequence), as in our case, measures that are highly sensitive to sequencing should be preferred. Good choices are the OM based algorithms. Shoval and Isaacson (2007) discussed the results from different algorithms and showed that OM can provide more accurate results when the difference comes from the sequencing.

Table D1. Common distance measurement.

Categories	Representative measure	Key parameter
Distances between probability distribution	CHI2(Deville and Saporta 1983) EUCLID (Gower 1982)	Sequence period for Chi-square n/a
Distance based on counts of common attributes	NMS (Elzinga 2003) SVRspell (Elzinga and Studer 2015)	n/a Length weights
Optimal matching	OM (Sankoff and Kruskal 1983) OMloc (Hollister 2009) OMspell (Halpin 2010) OMstran (Biemann 2011)	Substitution cost Expansion cost Expansion cost Transition cost and Indel cost

APPENDIX E. Summary of Hierarchical Clustering Algorithms

For creating the clusters, partitional (e.g., k-mean) and hierarchical clustering are the most commonly used techniques in social science (Reddy and Vinzamuri 2013). We recommend hierarchical clustering over partitional clustering for two reasons. First, partitional clustering requires a researcher to firstly assume and specify the number of partitions (e.g., k clusters), which is often impractical when trying to identify latent patterns; hierarchical clustering builds clusters incrementally without such a presumption and generates a dendrogram that depicts the hierarchical relationships of the sub-sequences in the form of a tree diagram. Therefore, it gives researchers a better overall picture of how sequences are related to and different from others, leaving room for the researcher to interpret the result through navigating the tree and deciding the best way to define clusters. This is especially beneficial for qualitative researchers with a critical realism view. Second, Mihaescu et al. (2009) showed that agglomerative hierarchical clustering is generally less sensitive to variations of the distance matrix that is created based on different substitution costs. This is especially advantageous for social science, where the accurate substitution cost is often unknown or not available. Lastly, the hierarchical approach tends to show more "deterministic" clustering results compared to the partitional clustering, as no random initialization is required (Reddy and Vinzamuri 2013). This also helps researchers to obtain consistent and reproducible results.

A major drawback of hierarchical clustering, compared with partitional clustering such as the *k*-mean, lies in efficiency, as the complexity of the *k*-mean algorithm is $O(NKd)$ for time and $O(N+K)$ for space. Hierarchical clustering is $O(N^2)$ for both time and space (Xu and Wunsch 2005). However, since frequent sub-sequences are typically much shorter than the entire sequence, this is generally not a serious issue for our framework.

Hierarchical methods can be classified into agglomerative (bottom-up) and divisive (top-down) clustering methods. Agglomerative clustering starts with every single object representing a single cluster, i.e., singleton at the bottom level, and continues to merge the two closest clusters into a single cluster at each step. Divisive clustering begins with one cluster containing all objects; at each stage, an existing cluster is divided into two. This will proceed recursively until singleton leave is obtained. However, algorithms that find the globally optimal division are computationally and conceptually very demanding, especially for the large dataset, so this divisive clustering is generally disregarded in most cases even though there is some experimental evidence that divisive algorithms produce more accurate hierarchies than agglomerative ones in some cases (Sharma et al. 2017).

Hierarchical clustering can be performed by most analytic or statistical tools, including commercial software, such as in SAS, SPSS, and MATLAB, as well as open source libraries, such as *cluster* and *kmer* for R, and *scikit-learn* for Python.

APPENDIX F. Clusters of sequences[27]

Cluster	Sequences
1	(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(CRED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(CRED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(CRED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(CRED)-(CRED)-(NCED)-(NCED)-(NCED)
1	(CRED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(CRED)
1	(NCED)-(NCED)-(NCED)-(NCTK)
1	(NCTK)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)
1	(NCED)-(NCED)-(NCED)-(CRED)
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)

[27] TraMineR also provides a cost method called “TRATE” that computes transition rates based on the data. However, the transition rate is calculated based on the probability from previous event to next event and is asymmetric (A->B can be different from B->A). For optimal matching, the substitution cost should be symmetric, as it describes the difference between two events (A-B). Therefore, use of TRATE will give unreliable results. We compare results based on TRATE and constant cost, and the results using constant cost are more reasonable to human interpretation as well.

1	(NCED)-(NCED)-(CRED)-(CRED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)
1	(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(CRED)
1	(NCED)-(NCED)-(CRED)-(CRED)-(CRED)
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCTK)
1	(NCTK)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(CRED)-(CRED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(CRED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(CRED)
1	(CRED)-(CRED)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCTK)
1	(NCED)-(NCED)-(NCED)-(CRED)-(CRED)-(CRED)
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)-(NCED)

1	(NCED)-(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(NCTK)-(NCED)-(NCED)-(NCED)
1	(NCTK)-(NCTK)-(NCED)-(NCED)-(NCED)
1	(NCED)-(NCED)-(NCED)-(NCED)-(CRED)-(CRED)
2	(CRED)-(NCED)-(NCED)
2	(CRED)-(CRED)-(NCED)
2	(CRED)-(CRED)-(CRED)
2	(CRED)-(CRED)-(CRED)-(NCED)
2	(CRED)-(CRED)-(NCED)-(NCED)
2	(CRED)-(CRED)-(CRED)-(NCED)-(NCED)
2	(NCTK)-(NCTK)-(NCTK)
2	(NCED)-(NCED)-(NCTK)
2	(NCED)-(NCTK)-(NCED)
2	(NCTK)-(NCTK)-(NCTK)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)
2	(CRED)-(CRED)-(CRTK)
2	(NCED)-(NCTK)-(NCTK)
2	(CRED)-(NCED)-(CRED)
2	(NCED)-(NCED)-(NCTK)-(NCED)
2	(NCTK)-(NCTK)-(NCED)
2	(CRED)-(CRTK)-(CRED)
2	(CRED)-(CRED)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(CRTK)
2	(NCED)-(CRED)-(NCED)
2	(CRED)-(CRED)-(NCED)-(CRED)
2	(NCED)-(NCED)-(NCTK)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(NCED)-(CRED)

2	(CRED)-(CRED)-(CRTK)-(CRED)
2	(NCTK)-(CRED)-(CRED)
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(NCED)
2	(NCED)-(NCTK)-(NCTK)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)
2	(NCED)-(CRED)-(NCED)-(NCED)
2	(CRED)-(NCED)-(NCTK)
2	(CRED)-(NCTK)-(NCED)
2	(CRED)-(CRED)-(CRED)-(NCED)-(NCED)-(NCED)
2	(NCTK)-(NCED)-(NCTK)
2	(NCTK)-(CRED)-(CRED)-(CRED)
2	(CRED)-(NCTK)-(CRED)
2	(NCED)-(NCTK)-(NCTK)-(NCED)
2	(NCTK)-(NCED)-(CRED)
2	(CRED)-(NCED)-(NCED)-(CRED)
2	(CRED)-(CRED)-(NCTK)-(NCED)
2	(NCED)-(NCTK)-(CRED)
2	(CRED)-(NCTK)-(NCTK)
2	(CRTK)-(NCTK)-(NCTK)
2	(NCED)-(NCTK)-(NCED)-(NCTK)
2	(CRED)-(CRED)-(NCTK)-(CRED)
2	(CRED)-(NCTK)-(CRED)-(CRED)
2	(CRED)-(CRTK)-(CRTK)
2	(NCTK)-(NCED)-(NCED)-(NCTK)
2	(CRED)-(CRED)-(CRED)-(NCTK)-(NCED)
3	(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)

3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(NCED)-(CRED)-(CRED)
3	(NCED)-(CRED)-(CRED)-(CRED)
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRTK)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(NCED)-(CRED)-(CRED)
3	(CRTK)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)
3	(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRTK)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)
3	(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(NCED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(NCED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)
3	(NCED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)
3	(CRED)-(CRED)-(CRED)-(CRED)-(NCTK)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(NCED)
3	(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)
3	(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCTK)
3	(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)

3	(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)
3	(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(NCTK)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)
3	(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCTK)
3	(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)
3	(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)
3	(NCED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(NCTK)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCTK)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(NCED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)

3	(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(NCED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)-(CRED)
3	(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)-(NCED)-(CRED)
3	(CRED)-(CRTK)-(CRED)-(CRED)-(CRED)-(CRED)-(CRED)

APPENDIX G. Example Retroductive Reasonings through Existing Theories

For instance, Van de Ven and Poole (1995) employed the substance metaphysics perspective of process theories to provide four motors or processes that can be used to establish a basic form of underlying generative mechanisms: life cycle, evolution, dialectic, and teleology. The life cycle model represents a process as development through different stages. Identifying sub-sequences from sequential pattern mining and their changes over time can help to identify this kind of process. For example, where the same sub-sequences appear mostly in certain periods (such as early phase, middle phase, or late phase) of different processes, this might suggest that the underlying mechanism follows a life cycle model or a teleological model. Therefore, researchers are advised to look at the nature of the sub-sequences and how changes forced upon them are driven by. If they are constructed by the actors' purposeful actions to achieve a goal, and changes are driven by external factors to reformulate, implement, and evaluate the goal, a teleological model is used to identify the underlying mechanism. On the other hand, if the sub-sequences are derived from prescribed logic or rules that drive a process to go through predefined stages in a sequence, a life-cycle model is used. When a period of a large number of variant primitive patterns is followed by a consolidation, which is followed by another period of a large number of variants, a researcher can posit a possible evolutionary model as the underlying generative mechanism. Thus, in the case of the biological evolutionary model of variation-selection-retention, the theoretical focus is then on how certain primitive patterns are selected and retained and how the results of the retention are manifested in the subsequent stages of the evolution. Finally, frequent and ongoing alternation of two primitive patterns, followed by a third primitive pattern might suggest an underlying mechanism that follows a dialectic model. In this case, the researchers may want to examine the two sets of sub-sequences and see if they are, in fact, representing opposing social forces. If the alternation does not converge into a third form, one can interpret it as a dialogical

relationship between two social forces which, although not covered by Van de Ven and Poole (1995), is another important form (Hargrave and Van De Ven 2006).

REFERENCES USED IN APPENDICES

Agrawal, R., and Srikant, R. 1995. "Mining Sequential Patterns," IEEE, pp. 3-14.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. 2002. "Sequential Pattern Mining Using a Bitmap Representation," *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*: ACM, pp. 429-435.

Biemann, T. 2011. "A Transition-Oriented Approach to Optimal Matching," *Sociological Methodology* (41:1), pp. 195-221.

Cornwell, B. (2015). *Social sequence analysis: Methods and applications* (Vol. 37). Cambridge University Press.

Deville, J.-C., and Saporta, G. 1983. "Correspondence Analysis, with an Extension Towards Nominal Time Series," *Journal of Econometrics* (22:1-2), pp. 169-189.

Elzinga, C. H. 2006. "Sequence Analysis: Metric Representations of Categorical Time Series," *Working Paper, available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.514.7995&rep=rep1&type=pdf*.

Elzinga, C. H., and Studer, M. 2015. "Spell Sequences, State Proximities, and Distance Metrics," *Sociological Methods & Research* (44:1), pp. 3-47.

Fournier-Viger, P., Gomariz, A., Campos, M. and Thomas, R., 2014, May. Fast vertical mining of sequential patterns using co-occurrence information. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 40-52). Springer, Cham.

Garofalakis, M. N., Rastogi, R., and Shim, K. 1999. "Spirit: Sequential Pattern Mining with Regular Expression Constraints," *VLDB*, pp. 7-10.

Gauthier, J.-A., Widmer, E. D., Bucher, P., and Notredame, C. 2009. "How Much Does It Cost? Optimization of Costs in Sequence Analysis of Social Science Data," *Sociological Methods & Research* (38:1), pp. 197-231.

Gower, J. C. 1982. "Euclidean Distance Geometry," *Math. Scientist* (7), pp 1 - 14.

Halpin, B. 2010. "Optimal Matching Analysis and Life-Course Data: The Importance of Duration," *Sociological Methods & Research* (38:3), pp. 365-388.

Hamming, R. W. 1950. "Error Detecting and Error Correcting Codes," *Bell Labs Technical Journal* (29:2), pp. 147-160.

Hargrave, T. J., and Van De Ven, A. H. 2006. "A Collective Action Model of Institutional Innovation," *Academy of Management Review* (31:4), pp. 864-888.

Han, J., Pei, J., and Yin, Y. 2000. "Mining Frequent Patterns without Candidate Generation," *ACM Sigmod Record*: ACM, pp. 1-12.

Hollister, M. 2009. "Is Optimal Matching Suboptimal?," *Sociological Methods & Research* (38:2), pp. 235-264.

Levenshtein, V. 1966. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, p. 707.

Li, Y., and Liu, B. 2007. "A Normalized Levenshtein Distance Metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (29:6), pp. 1091-1095.

Masseglia, F., Cathala, F., and Poncelet, P. 1998. "The PSP Approach for Mining Sequential Patterns," *European Symposium on Principles of Data Mining and Knowledge Discovery*: Springer, pp. 176-184.

Mihaescu, R., Levy, D., and Pachter, L. 2009. "Why Neighbor-Joining Works," *Algorithmica* (54), pp. 1-24.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., ... & Hsu, M. C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering,* 16(11), pp. 1424-1440.

Reddy, C.K. and Vinzamuri, B., 2013. A Survey of Partitional and Hierarchical Clustering Algorithms. Data clustering: Algorithms and applications, 87.

Sankoff, D., and Kruskal, J. 1983. "*Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*," Addison-Wesley, Reading, MA.

Shoval, N., and Isaacson, M. 2007. "Sequence Alignment as a Method for Human Activity Analysis in Space and Time," *Annals of the Association of American Geographers* (97), pp. 282-297.

Srikant, R., and Agrawal, R. 1996. "Mining Sequential Patterns: Generalizations and Performance Improvements," *International Conference on Extending Database Technology*: Springer, pp. 1-17.

Studer, M., and Ritschard, G. 2014. "A Comparative Review of Sequence Dissimilarity Measures," Working Paper, DOI : 10.12682/lives. 2296-1658.2014.33, available at *https://archive-ouverte.unige.ch/unige:78575*.

Studer, M., and Ritschard, G. 2016. "What Matters in Differences between Life Trajectories: A Comparative Review of Sequence Dissimilarity Measures," *Journal of the Royal Statistical Society: Series A (Statistics in Society)* (179:2), pp. 481-511.

Van de Ven, A. H., and Poole, M. S. 1995. "Explaining Development and Change in Organizations," *Academy of Management Review* (20), pp. 510-540.

Wilson, C. 2006. "Reliability of Sequence-Alignment Analysis of Social Processes: Monte Carlo Tests of Clustalg Software," *Environment and Planning A* (38), pp. 187-204.

Xu, D. and Tian, Y., 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2), pp. 165-193.

Xu, R., and Wunsch, D. 2005. "Survey of Clustering Algorithms," *IEEE Transactions on Neural Networks* (16:3), pp. 645-678.

Yang, Z., Wang, Y., and Kitsuregawa, M. 2007. "Lapin: Effective Sequential Pattern Mining Algorithms by Last Position Induction for Dense Databases," *International Conference on Database Systems for Advanced Applications*: Springer, pp. 1020-1023.

Zaki, M.J., 2001. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, (42:1-2), pp. 31-60.

Zhang, M., Kao, C., Yip, C., and Cheung, D. 2001. "A GSP-Based Efficient Algorithm for Mining Frequent Sequences," *Proceedings of 2001 International Conference on Artificial Intelligence (IC-AI 2001)*.